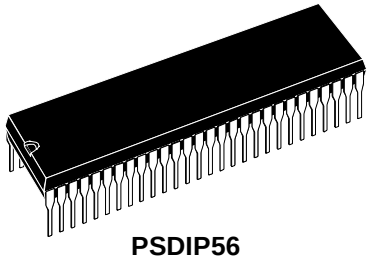
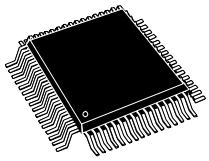
ST92195C/D

48-96 Kbyte ROM HCMOS MCU WITH ON-SCREEN DISPLAY AND TELETEXT DATA SLICER

- Register File based 8/16 bit Core Architecture with RUN, WFI, SLOW and HALT modes
- 0°C to +70°C operating temperature range
- Up to 24 MHz. operation @ 5V±10%
- Min. instruction cycle time: 165ns at 24 MHz.
- 48, 56, 64, 84 or 96 Kbytes ROM
- 256 bytes RAM of Register file (accumulators or index registers)
- 256 to 512 bytes of on-chip static RAM
- 2 or 8 Kbytes of TDSRAM (Teletext and Display Storage RAM)
- 28 fully programmable I/O pins
- Serial Peripheral Interface
- Flexible Clock controller for OSD, Data Slicer and Core clocks running from a single low frequency external crystal.
- Enhanced display controller with 26 rows of 40/80 characters
  - 2 sets of 512 characters
  - Serial and Parallel attributes
  - 10x10 dot matrix, definable by user
  - 4/3 and 16/9 supported in 50/60Hz and 100/120 Hz mode
  - Rounding, fringe, double width, double height, scrolling, cursor, full background color, half-intensity color, translucency and half-tone modes
- Teletext unit, including Data Slicer, Acquisition Unit and up to 8 Kbytes RAM for data storage
- VPS and Wide Screen Signalling slicer
- Integrated Sync Extractor and Sync Controller
- 14-bit Voltage Synthesis for tuning reference voltage
- Up to 6 external interrupts plus one Non-Maskable Interrupt
- 8 x 8-bit programmable PWM outputs with 5V open-drain or push-pull capability
- 16-bit watchdog timer with 8-bit prescaler
- 1 or 2 16-bit standard timer(s) with 8-bit prescaler



PSDIP56



TQFP64

See end of Datasheet for ordering information

- I²C Master/Slave (on some devices)
- 4-channel A/D converter; 5-bit guaranteed
- Rich instruction set and 14 addressing modes
- Versatile development tools, including Assembler, Linker, C-compiler, Archiver, Source Level Debugger and hardware emulators with Real-Time Operating System available from third parties
- Pin-compatible EPROM and OTP devices available

Device Summary

Device	ROM	RAM	TDSRAM	I²C	Timer
ST92195C3	48K	256	2K	No	1
ST92195C4		512	6K		
ST92195C5			8K		
ST92195C6	56K				
ST92195C7	64K				
ST92195C8	84K				
ST92195C9	96K				
ST92195D5	48K	512	8K	Yes	2
ST92195D6	56K				
ST92195D7	64K				

Table of Contents

ST92195C/D 1

1 GENERAL DESCRIPTION 7

1.1 INTRODUCTION 7

1.1.1 ST9+ Core 7
1.1.2 Power Saving Modes 7
1.1.3 I/O Ports 7
1.1.4 TV Peripherals 7
1.1.5 On Screen Display 7
1.1.6 Teletext and Display Storage RAM 7
1.1.7 Teletext, VPS and WSS Data Slicers 8
1.1.8 Voltage Synthesis Tuning Control 8
1.1.9 PWM Output 8
1.1.10 Serial Peripheral Interface (SPI) 8
1.1.11 Standard Timer (STIM) 8
1.1.12 I²C Bus Interface 8
1.1.13 Analog/Digital Converter (ADC) 8

1.2 PIN DESCRIPTION 10

1.2.1 I/O Port Alternate Functions. 14

1.3 MEMORY MAP 17

1.4 REGISTER MAP 18

2 DEVICE ARCHITECTURE 24

2.1 CORE ARCHITECTURE 24

2.2 MEMORY SPACES 24

2.2.1 Register File 24
2.2.2 Register Addressing 26

2.3 SYSTEM REGISTERS 27

2.3.1 Central Interrupt Control Register 27
2.3.2 Flag Register 28
2.3.3 Register Pointing Techniques 29
2.3.4 Paged Registers 32
2.3.5 Mode Register 32
2.3.6 Stack Pointers 33

2.4 MEMORY ORGANIZATION 35

2.5 MEMORY MANAGEMENT UNIT 36

2.6 ADDRESS SPACE EXTENSION 37

2.6.1 Addressing 16-Kbyte Pages 37
2.6.2 Addressing 64-Kbyte Segments 38

2.7 MMU REGISTERS 38

2.7.1 DPR[3:0]: Data Page Registers 38
2.7.2 CSR: Code Segment Register 40
2.7.3 ISR: Interrupt Segment Register 40
2.7.4 DMASR: DMA Segment Register 40

2.8 MMU USAGE 42

2.8.1 Normal Program Execution 42
2.8.2 Interrupts 42

Table of Contents

2.8.3 DMA 42
3 INTERRUPTS 43
3.1 INTRODUCTION 43
3.2 INTERRUPT VECTORING 43
3.2.1 Divide by Zero trap 43
3.2.2 Segment Paging During Interrupt Routines 44
3.3 INTERRUPT PRIORITY LEVELS 44
3.4 PRIORITY LEVEL ARBITRATION 44
3.4.1 Priority Level 7 (Lowest) 44
3.4.2 Maximum Depth of Nesting 44
3.4.3 Simultaneous Interrupts 45
3.4.4 Dynamic Priority Level Modification 45
3.5 ARBITRATION MODES 45
3.5.1 Concurrent Mode 45
3.5.2 Nested Mode 48
3.6 EXTERNAL INTERRUPTS 50
3.7 TOP LEVEL INTERRUPT 53
3.8 ON-CHIP PERIPHERAL INTERRUPTS 53
3.9 INTERRUPT RESPONSE TIME 54
3.10 INTERRUPT REGISTERS 55
4 RESET AND CLOCK CONTROL UNIT (RCCU) 59
4.1 INTRODUCTION 59
4.2 RESET / STOP MANAGER 59
4.3 OSCILLATOR CHARACTERISTICS 59
4.4 CLOCK CONTROL REGISTERS 61
4.5 RESET CONTROL UNIT REGISTERS 62
5 TIMING AND CLOCK CONTROLLER 63
5.1 FREQUENCY MULTIPLIERS 63
5.2 REGISTER DESCRIPTION 66
5.2.1 Register Mapping 67
6 I/O PORTS 68
6.1 INTRODUCTION 68
6.2 SPECIFIC PORT CONFIGURATIONS 68
6.3 PORT CONTROL REGISTERS 68
6.4 INPUT/OUTPUT BIT CONFIGURATION 69
6.5 ALTERNATE FUNCTION ARCHITECTURE 73
6.5.1 Pin Declared as I/O 73
6.5.2 Pin Declared as an Alternate Function Input 73
6.5.3 Pin Declared as an Alternate Function Output 73

Table of Contents

6.6 I/O STATUS AFTER WFI, HALT AND RESET 73

7 ON-CHIP PERIPHERALS 74

7.1 TIMER/WATCHDOG (WDT) 74

7.1.1 Introduction 74
7.1.2 Functional Description 75
7.1.3 Watchdog Timer Operation 76
7.1.4 WDT Interrupts 78
7.1.5 Register Description 79

7.2 STANDARD TIMER (STIM) 81

7.2.1 Introduction 81
7.2.2 Functional Description 82
7.2.3 Interrupt Selection 83
7.2.4 Register Mappingl 83
7.2.5 Register Description 84

7.3 TELETEXT DISPLAY STORAGE RAM INTERFACE 85

7.3.1 Introduction 85
7.3.2 Functional Description 86
7.3.3 Initialisation 89
7.3.4 Register Description 90

7.4 ON SCREEN DISPLAY (OSD) 93

7.4.1 Introduction 93
7.4.2 General Features 93
7.4.3 Functional Description 96
7.4.4 Programming the Display 99
7.4.5 Vertical Scrolling Control 113
7.4.6 Display Memory Mapping Examples 114
7.4.7 Font Mapping 120
7.4.8 Font Mapping Modes 120
7.4.9 Register Description 124
7.4.10 Application Software Examples 136

7.5 SYNC CONTROLLER 139

7.5.1 H/V Polarity Control 140
7.5.2 Field Detection 140
7.5.3 Interrupt Generation 140
7.5.4 Sync Controller Working Modes 140
7.5.5 Register Description 143

7.6 SYNC EXTRACTOR 145

7.6.1 Time Windowing For Slicers 145
7.6.2 Field Detection 145
7.6.3 CVBS Amplitude Information 145
7.6.4 CVBS Signal Quality 145
7.6.5 Sync Extractor Switch Off 145

7.7 TELETEXT SLICER AND ACQUISITION UNIT 146

7.7.1 Introduction 146
7.7.2 Main Features 146
7.7.3 Teletext Slicer 147
7.7.4 Acquisition Unit 148

Table of Contents

7.7.5 Hamming Decoding 154
7.7.6 Teletext Signal Quality Measure 156
7.7.7 Register Description 157
7.8 VPS & WSS SLICER 164
7.8.1 Introduction 164
7.8.2 General Operation 165
7.8.3 About Video Programming System (VPS) 166
7.8.4 About Wide-Screen-Signaling (WSS) 167
7.8.5 WSS Signal Content 167
7.8.6 WSS Data Group Assignments 168
7.8.7 Register Description 169
7.9 SERIAL PERIPHERAL INTERFACE (SPI) 173
7.9.1 Introduction 173
7.9.2 Device-Specific Options 173
7.9.3 Functional Description 174
7.9.4 Interrupt Structure 175
7.9.5 Working With Other Protocols 176
7.9.6 I2C-bus Interface 176
7.9.7 S-Bus Interface 179
7.9.8 IM-bus Interface 180
7.9.9 Register Description 181
7.10 TWO-CHANNEL I^2C BUS INTERFACE (I2C) 183
7.10.1 Introduction 183
7.10.2 General Description 184
7.10.3 Functional Description 185
7.10.4 Interrupt Handling 187
7.10.5 Error Cases 187
7.10.6 Register Description 188
7.11 A/D CONVERTER (A/D) 195
7.11.1 Introduction 195
7.11.2 Main Features 195
7.11.3 General Description 195
7.11.4 Register Description 197
7.12 VOLTAGE SYNTHESIS TUNING CONVERTER (VS) 199
7.12.1 Description 199
7.12.2 Output Waveforms 199
7.12.3 Register Description 203
7.13 PWM GENERATOR 204
7.13.1 Introduction 204
7.13.2 Register Mapping 205
8 ELECTRICAL CHARACTERISTICS 209
9 GENERAL INFORMATION 215
9.1 PACKAGE MECHANICAL DATA 215
9.2 ORDERING INFORMATION 216
9.2.1 Transfer Of Customer Code 216

Table of Contents

ST92E195C/D-ST92T195C/D 217

1 GENERAL DESCRIPTION 219

1.0.1 ST9+ Core 219
1.0.2 Power Saving Modes 219
1.0.3 I/O Ports 219
1.0.4 TV Peripherals 219
1.0.5 On Screen Display 219
1.0.6 Teletext and Display Storage RAM 219
1.0.7 Teletext, VPS and WSS Data Slicers 220
1.0.8 Voltage Synthesis Tuning Control 220
1.0.9 PWM Output 220
1.0.10 Serial Peripheral Interface (SPI) 220
1.0.11 Standard Timer (STIM) 220
1.0.12 I²C Bus Interface 220
1.0.13 Analog/Digital Converter (ADC) 220

1.1 PIN DESCRIPTION 222

1.1.1 I/O Port Alternate Functions. 226
1.1.2 I/O Port Styles 227

1.2 MEMORY MAP 229

1.3 REGISTER MAP 230

2 ELECTRICAL CHARACTERISTICS 236

3 GENERAL INFORMATION 243

3.1 PACKAGE MECHANICAL DATA 243

3.2 ORDERING INFORMATION 245

3.2.1 Transfer Of OSD Customer Code 245

4 REVISION HISTORY 247

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST92195C and ST92195D microcontrollers are developed and manufactured by STMicroelectronics using a proprietary n-well HCMOS process. Their performance derives from the use of a flexible 256-register programming model for ultra-fast context switching and real-time event response. The intelligent on-chip peripherals offload the ST9 core from I/O and data management processing tasks allowing critical application tasks to get the maximum use of core resources. The ST92195C/D MCU support low power consumption and low voltage operation for power-efficient and low-cost embedded systems.

1.1.1 ST9+ Core

The advanced Core consists of the Central Processing Unit (CPU), the Register File and the Interrupt controller.

The general-purpose registers can be used as accumulators, index registers, or address pointers. Adjacent register pairs make up 16-bit registers for addressing or 16-bit processing. Although the ST9 has an 8-bit ALU, the chip handles 16-bit operations, including arithmetic, loads/stores, and memory/register and memory/memory exchanges.

Two basic addressable spaces are available: the Memory space and the Register File, which includes the control and status registers of the on-chip peripherals.

1.1.2 Power Saving Modes

To optimize performance versus power consumption, a range of operating modes can be dynamically selected.

Run Mode. This is the full speed execution mode with CPU and peripherals running at the maximum clock speed delivered by the Phase Locked Loop (PLL) of the Clock Control Unit (CCU).

Wait For Interrupt Mode. The Wait For Interrupt (WFI) instruction suspends program execution until an interrupt request is acknowledged. During WFI, the CPU clock is halted while the peripherals and interrupt controller keep running at a frequency programmable via the CCU. In this mode, the power consumption of the device can be reduced by more than 95% (Low power WFI).

Halt Mode. When executing the HALT instruction, and if the Watchdog is not enabled, the CPU and its peripherals stop operating and the status of the machine remains frozen (the clock is also stopped). A reset is necessary to exit from Halt mode.

1.1.3 I/O Ports

Up to 28 I/O lines are dedicated to digital Input/ Output. These lines are grouped into up to five I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer and output, analog inputs, external interrupts and serial or parallel I/O.

1.1.4 TV Peripherals

A set of on-chip peripherals form a complete system for TV set and VCR applications:

– Voltage Synthesis

– VPS/WSS Slicer

– Teletext Slicer

– Teletext Display RAM

– OSD

1.1.5 On Screen Display

The human interface is provided by the On Screen Display module, this can produce up to 26 lines of up to 80 characters from a ROM of two 512-character sets. The character resolution is 10x10 dot. Four character sizes are supported. Serial attributes allow the user to select foreground and background colors, character size and fringe background. Parallel attributes can be used to select additional foreground and background colors and underline on a character by character basis.

1.1.6 Teletext and Display Storage RAM

The internal Teletext and Display storage RAM can be used to store Teletext pages as well as Display parameters.

INTRODUCTION (Cont'd)

1.1.7 Teletext, VPS and WSS Data Slicers

The three on-board data slicers using a single external crystal are used to extract the Teletext, VPS and WSS information from the video signal. Hardware Hamming decoding is provided.

1.1.8 Voltage Synthesis Tuning Control

14-bit Voltage Synthesis using the PWM (Pulse Width Modulation)/BRM (Bit Rate Modulation) technique can be used to generate tuning voltages for TV set applications. The tuning voltage is output on one of two separate output pins.

1.1.9 PWM Output

Control of TV settings can be made with up to eight 8-bit PWM outputs, with a maximum frequency of 23,437Hz at 8-bit resolution (INTCLK = 12 MHz). Low resolutions with higher frequency operation can be programmed.

1.1.10 Serial Peripheral Interface (SPI)

The SPI bus is used to communicate with external devices via the SPI, or I²C bus communication standards. The SPI uses a single data line for data input and output. A second line is used for a synchronous clock signal.

1.1.11 Standard Timer (STIM)

The ST92195C and ST92195D have one or two Standard Timer(s) that include a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes.

1.1.12 I²C Bus Interface

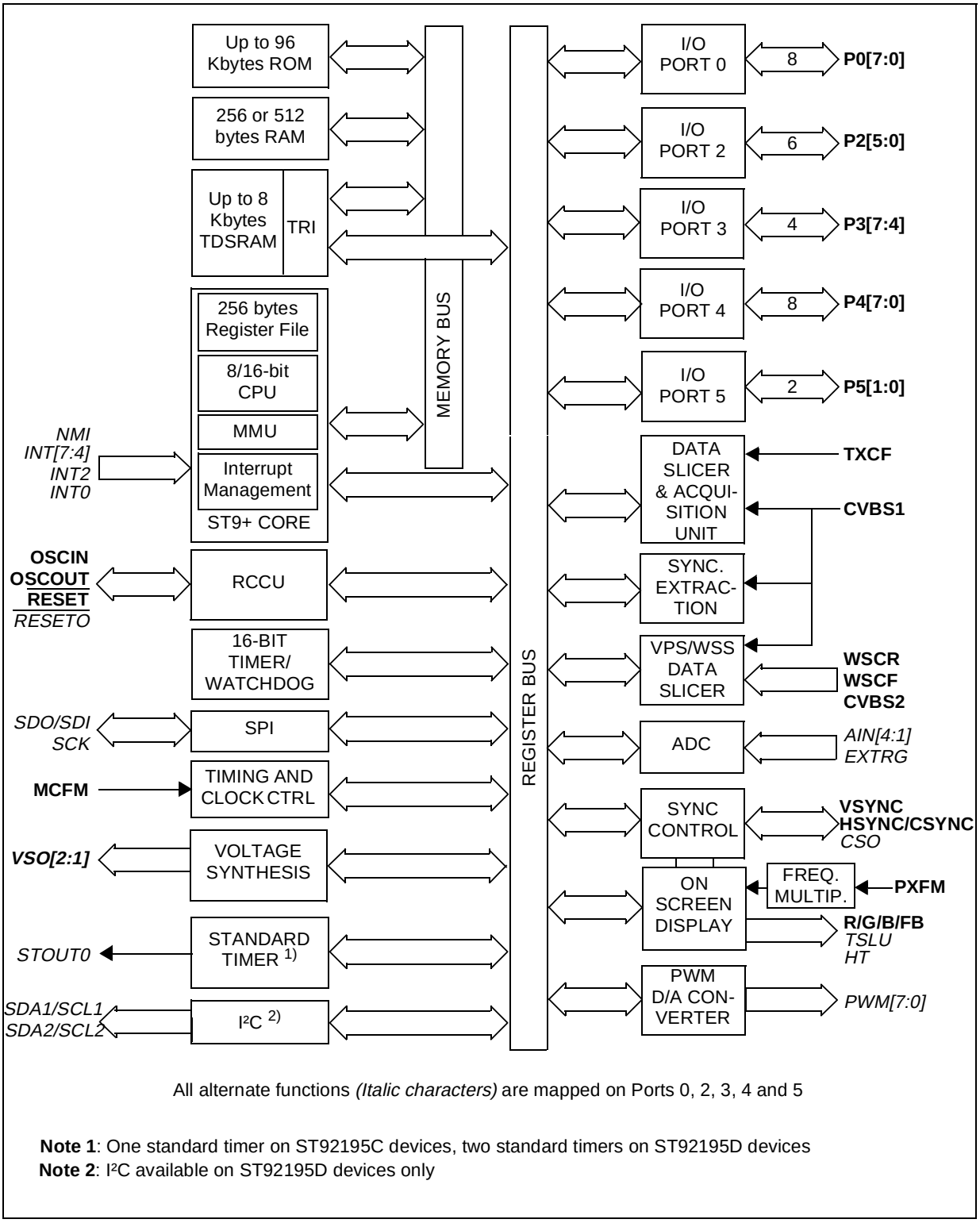
The ST92195D versions have one I²C bus interface. The I²C bus is a synchronous serial bus for connecting multiple devices using a data line and a clock line. Multimaster and slave modes are supported. Up to two channels are supported. The I²C interface supports 7-bit addressing. It supports speeds of up to 800 KHz. Bus events (Bus busy, slave address recognised) and error conditions are automatically flagged in peripheral registers and interrupts are optionally generated.

1.1.13 Analog/Digital Converter (ADC)

In addition there is a 4-channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

INTRODUCTION (Cont'd)

Figure 1. ST92195C/D Block Diagram

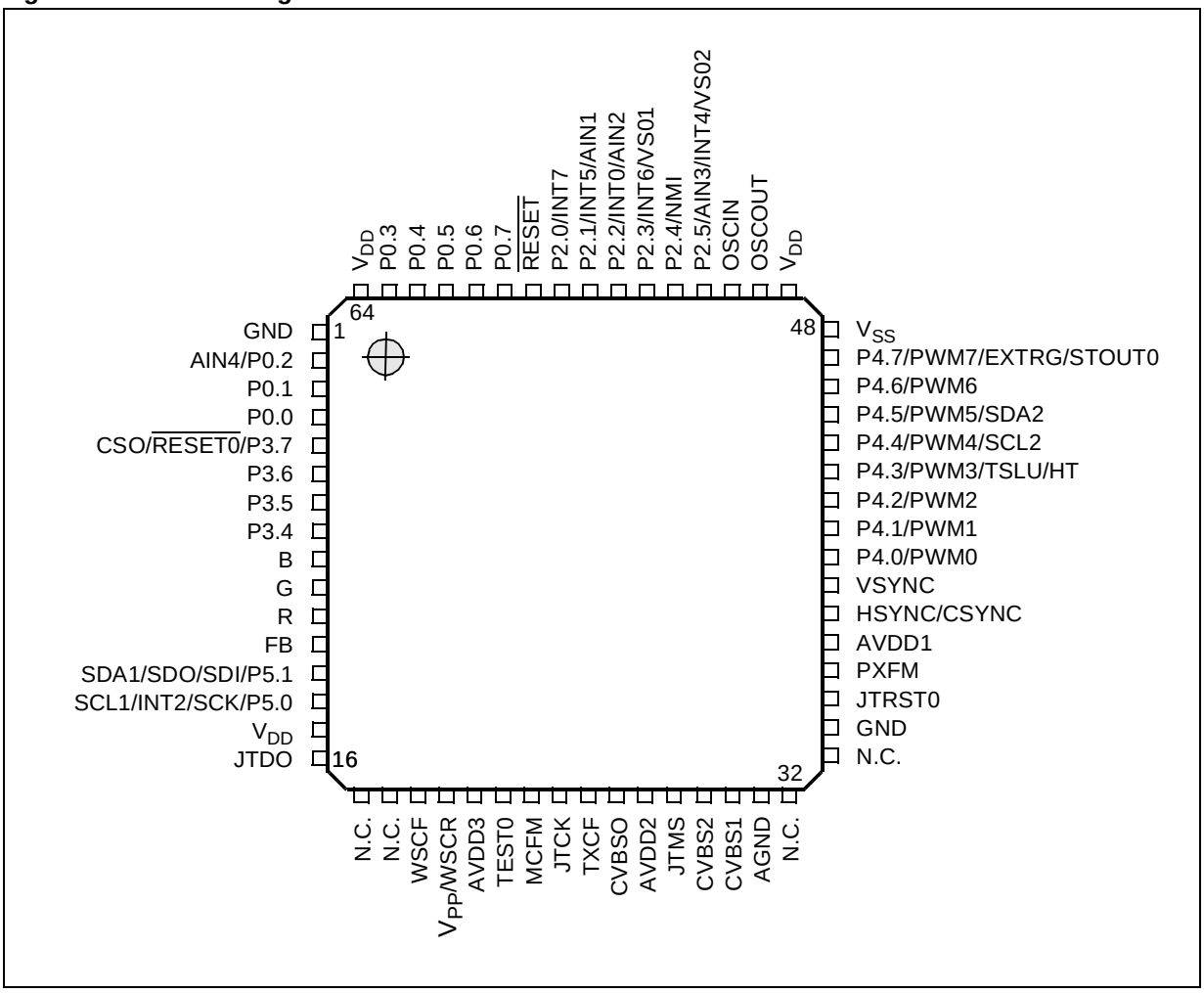


All alternate functions *(Italic characters)* are mapped on Ports 0, 2, 3, 4 and 5

Note 1: One standard timer on ST92195C devices, two standard timers on ST92195D devices
Note 2: I²C available on ST92195D devices only

1.2 PIN DESCRIPTION

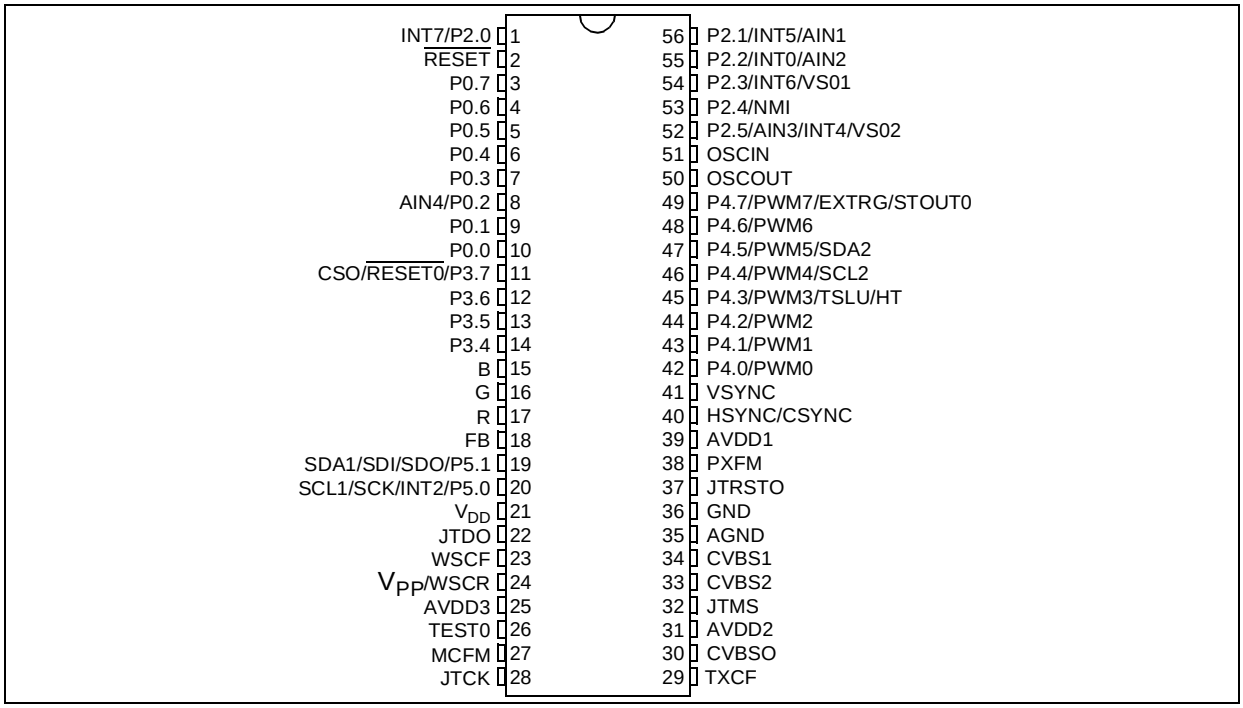
Figure 2. 64-Pin Package Pin-Out



N.C. = Not connected

PIN DESCRIPTION (Cont'd)

Figure 3. 56-Pin Package Pin-Out



RESET *Reset* (input, active low). The ST9+ is initialised by the Reset signal. With the deactivation of RESET, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

R/G/B *Red/Green/Blue*. Video color analog DAC outputs.

FB *Fast Blanking*. Video analog DAC output.

V_{DD} Main power supply voltage (5V±10%, digital)

WSCF, WSCR Analog pins for the VPS/WSS slicer . These pins must be tied to ground or not connected.

V_{PP}: On EPROM/OTP devices, the WSCR pin is replaced by V_{PP} which is the programming voltage pin. V_{PP} should be tied to GND in user mode.

MCFM Analog pin for the display pixel frequency multiplier.

OSCIN, OSCOUT *Oscillator* (input and output). These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

VSYNC *Vertical Sync*. Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC/CSYNC *Horizontal/Composite sync*. Horizontal or composite video synchronisation input to OSD. Positive or negative polarity.

PXFM Analog pin for the Display Pixel Frequency Multiplier

AVDD3 *Analog V_{DD} of PLL.* This pin must be tied to V_{DD} externally.

GND Digital circuit ground.

AGND Analog circuit ground (must be tied externally to digital GND).

CVBS1 Composite video input signal for the Teletext slicer and sync extraction.

CVBS2 Composite video input signal for the VPS/WSS slicer. Pin AC coupled.

AVDD1, AVDD2 Analog power supplies (must be tied externally to AVDD3).

TXCF Analog pin for the Teletext slicer line PLL.

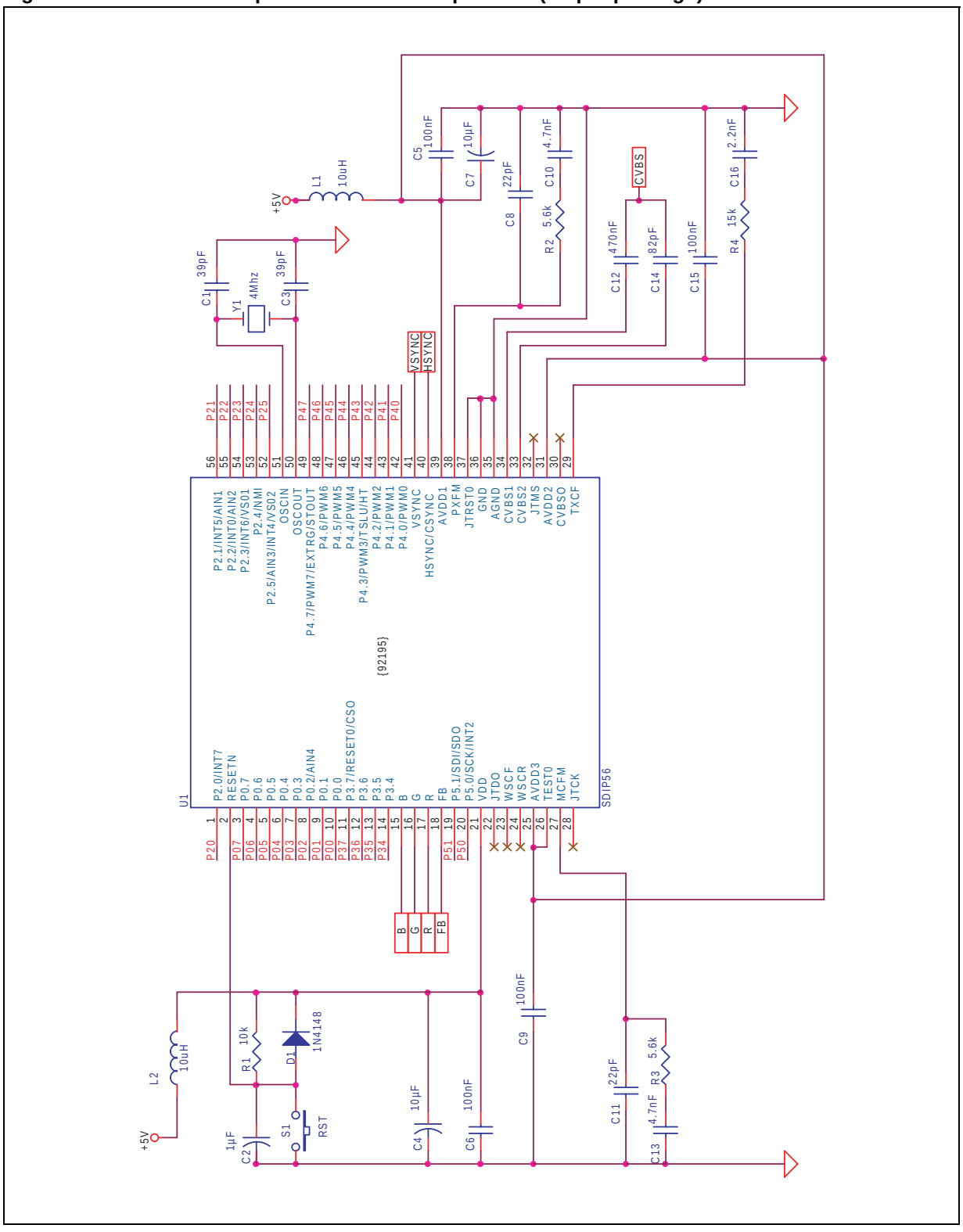
CVBSO, JTDO, JTCK Test pins: leave floating.

TEST0 Test pins: must be tied to AVDD2.

JTRST0 Test pin: must be tied to GND.

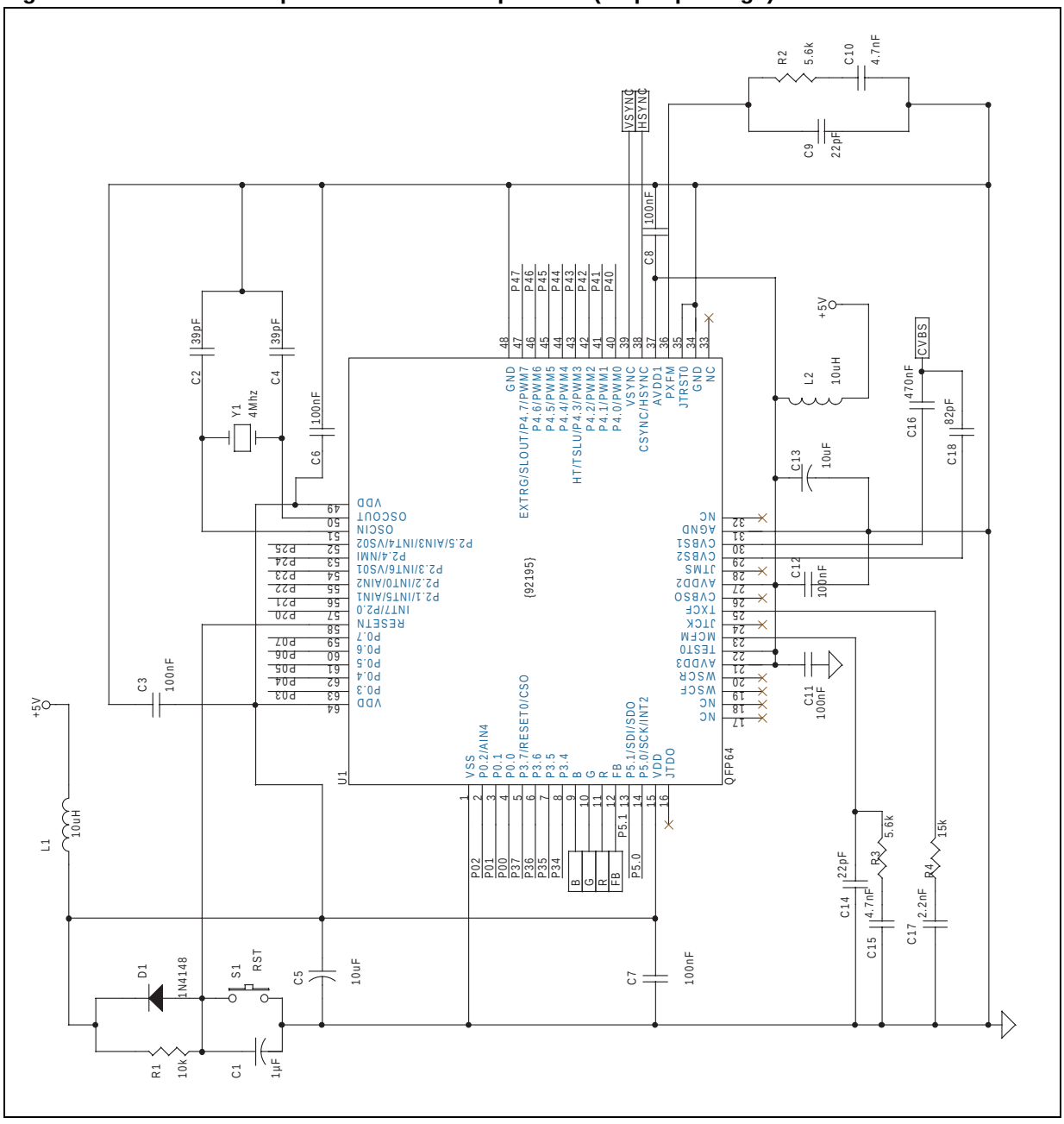
PIN DESCRIPTION (Cont'd)

Figure 4. ST92195C/D Required External Components (56-pin package)



PIN DESCRIPTION (Cont'd)

Figure 5. ST92195C/D Required External Components (64-pin package)



PIN DESCRIPTION (Cont'd)

P0[7:0], P2[5:0], P3[7:4], P4[7:0], P5[1:0]
I/O Port Lines (Input/Output, TTL or CMOS compatible).

28 lines grouped into I/O ports, bit programmable as general purpose I/O or as Alternate functions (see I/O section).

Important: Note that open-drain outputs are for logic levels only and are not true open drain.

1.2.1 I/O Port Alternate Functions.

Each pin of the I/O ports of the ST92195C/D may assume software programmable Alternate Functions (see Table 1).

Table 1. ST92195C/D I/O Port Alternate Function Summary

Port Name	General Purpose I/O	Pin No.		Alternate Functions		
		TQFP64	SDIP56			
P0.0	All ports useable for general purpose I/O (input, output or bidirectional)	4	10		I/O	
P0.1		3	9		I/O	
P0.2		2	8	AIN4	I	A/D Analog Data Input 4
P0.3		63	7		I/O	
P0.4		62	6		I/O	
P0.5		61	5		I/O	
P0.6		60	4		I/O	
P0.7		59	3		I/O	
P2.0		57	1	INT7	I	External Interrupt 7
P2.1		56	56	AIN1	I	A/D Analog Data Input 1
				INT5	I	External Interrupt 5
P2.2		55	55	INT0	I	External Interrupt 0
				AIN2	I	A/D Analog Data Input 2
P2.3		54	54	INT6	I	External Interrupt 6
				VSO1	O	Voltage Synthesis Output 1
P2.4		53	53	NMI	I	Non Maskable Interrupt Input
P2.5		52	52	AIN3	I	A/D Analog Data Input 3
				INT4	I	External Interrupt 4
				VSO2	O	Voltage Synthesis Output 2
P3.4		8	14		I/O	
P3.5		7	13		I/O	
P3.6		6	12		I/O	
P3.7		5	11	$\overline{\text{RESET0}}$	O	Internal Reset Output
				CSO	O	Composite Sync output
P4.0		40	42	PWM0	O	PWM Output 0
P4.1		41	43	PWM1	O	PWM Output 1
P4.2		42	44	PWM2	O	PWM Output 2
P4.3		43	45	PWM3	O	PWM Output 3
				TSLU	O	Translucency Digital Output
				HT	O	Half-tone Output

Port Name	General Purpose I/O	Pin No.		Alternate Functions		
		TQFP64	SDIP56			
P4.4	All ports useable for general purpose I/O (input, output or bidirectional)	44	46	PWM4	O	PWM Output 4
				SCL2	I/O	I²C Channel 2 Serial Clock [1]
P4.5		45	47	PWM5	O	PWM Output 5
				SDA2	I/O	I²C Channel 2 Serial Data [1]
P4.6		46	48	PWM6	O	PWM Output 6
P4.7		47	49	EXTRG	I	A/D Converter External Trigger Input
				PWM7	O	PWM Output 7
				STOUT0	O	Standard Timer 0 Output
P5.0		14	20	INT2	I	External Interrupt 2
				SCK	O	SPI Serial Clock
				SCL1	I/O	I²C Channel 1 Serial Clock [1]
P5.1		13	19	SDO	O	SPI Serial Data Out
				SDI	I	SPI Serial Data In
				SDA1	I/O	I²C Channel 1 Serial Data [1]

Note 1: I²C available on ST92195D devices only.

Table 2. I/O Port Styles

Pins	Weak Pull-Up	Port Style	Reset Values
P0[7:0]	no	Standard I/O	BID / OD / TTL
P2[5,4,3,2]	no	Standard I/O	BID / OD / TTL
P2[1,0]	no	Schmitt trigger	BID / OD / TTL
P3.7	yes	Standard I/O	AF / PP / TTL
P3[6,5,4]	no	Standard I/O	BID / OD / TTL
P4[7:0]	no	Standard I/O	BID / OD / TTL
P5[1:0]	no	Standard I/O	BID / OD / TTL

Legend:

AF= Alternate Function, BID = Bidirectional, OD = Open Drain
PP = Push-Pull, TTL = TTL Standard Input Levels

How to Read this Table

To configure the I/O ports, use the information in this table and the Port Bit Configuration Table in the I/O Ports Chapter on page 71.

Port Style= the hardware characteristics fixed for each port line.

Inputs:

– If port style = Standard I/O, either TTL or CMOS input level can be selected by software.

– If port style = Schmitt trigger, selecting CMOS or TTL input by software has no effect, the input will always be Schmitt Trigger.

Weak Pull-Up = This column indicates if a weak pull-up is present or not.

– If WPU = yes, then the WPU can be enabled/disable by software

– If WPU = no, then enabling the WPU by software has no effect

Alternate Functions (AF) = More than one AF cannot be assigned to an external pin at the same time:

An alternate function can be selected as follows.

AF Inputs:

– AF is selected implicitly by enabling the corresponding peripheral. Exception to this are ADC analog inputs which must be explicitly selected as AF by software.

PIN DESCRIPTION (Cont'd)

AF Outputs or Bidirectional Lines:

– In the case of Outputs or I/Os, AF is selected explicitly by software.

Example 1: ADC trigger digital input

AF: EXTRG, Port: P4.7, Port Style: Standard I/O.

Write the port configuration bits (for TTL level):

P4C2.7=1

P4C1.7=0

P4C0.7=1

Enable the ADC trigger by software as described in the ADC chapter.

Example 2: PWM 0 output

AF: PWM0, Port: P4.0

Write the port configuration bits (for output push-pull):

P4C2.0=0

P4C1.0=1

P4C0.0=1

Example 3: ADC analog input

AF: AIN1, Port : P2.1, Port style: does not apply to analog inputs

Write the port configuration bits:

P2C2.1=1

P2C1.1=1

P2C0.1=1

1.3 MEMORY MAP

Internal ROM

The ROM memory is mapped in two segments: segment 00h and segment 01h; It starts at address 0000h in MMU segment 00h.

Device	Size	Start Address	End Address
ST92195C3/C4/C5 ST92195D5	48K	00 0000h	00 BFFFh
ST92195C6 ST92195D6	56K	00 0000h	00 DFFFh
ST92195C7 ST92195D7	64K	00 0000h	00 FFFFh
ST92195C8	84K	00 0000h	01 4FFFh
ST92195C9	96K	00 0000h	01 7FFFh

Internal RAM, 256 or 512 bytes

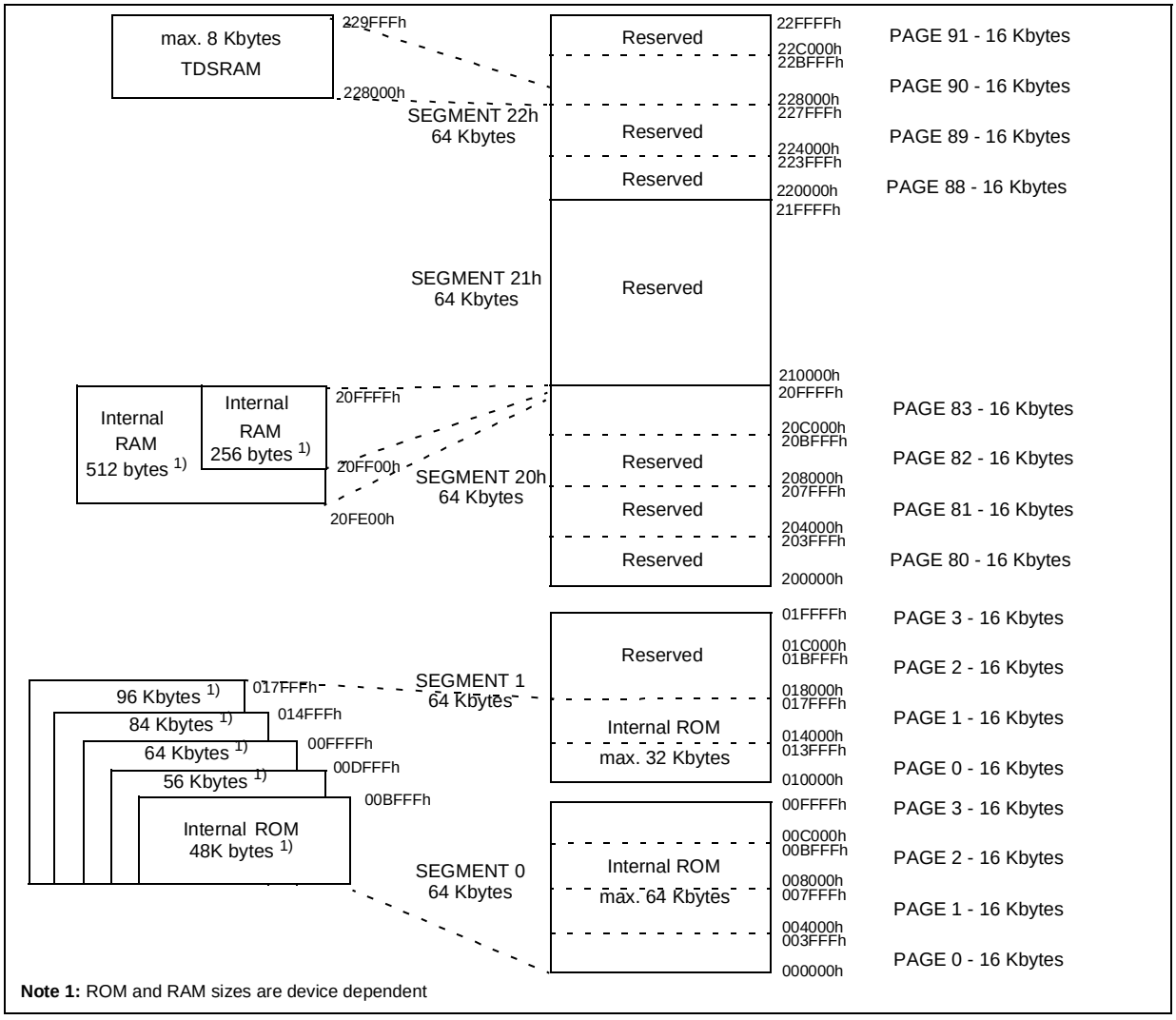
The internal RAM is mapped in MMU segment 20h, from address FF00h to FFFFh or from FE00h to FFFFh.

Internal TDSRAM

The Internal TDSRAM is mapped starting at address 8000h in MMU segment 22h. It is a fully static memory.

Device	Size	Start Address	End Address
ST92195C3	2K	8000h	87FFh
ST92195C4	6K	8000h	97FFh
ST92195C5/C6/C7/C8/C9 ST92195D5/D6/D7	8K	8000h	9FFFh

Figure 6. ST92195C/D Memory Map



Note 1: ROM and RAM sizes are device dependent

1.4 REGISTER MAP

The following pages contain a list of ST92195C/D registers, grouped by peripheral or function.

Be very careful to correctly program both:

– The set of registers dedicated to a particular function or peripheral.

– Registers common to other functions.

In particular, double-check that any registers with “undefined” reset values have been correctly initialised.

Warning: Note that in the **EIVR** and each **IVR** register, all bits are significant. Take care when defining base vector addresses that entries in the Interrupt Vector table do not overlap.

Table 3. Group F Pages Register Map

Register	Page																
	0	2	3	6	11	21	32	33	34	35	36	37	38	39	44	55	59	62
R255	Res.	Res.	Res.	VPS/ WSS	Res.	Res.	OSD		ACQ	TSU	TDS RAM			Res.	Res	RCCU (PLL)	VS	Res.
R254	SPI	Port 3												TCC				
R253																	Res.	
R252	WCR				Res.	MMU											PWM	
R251	WDT	Res.			STIM 1 1)									Res.		Res.		
R250		Port 2																
R249																		Res.
R248																		
R247	EXT INT	Res.	Res.		Res	Res.				Res.				Res.	I²C 1)	Res.		
R246			Port 5			MMU												
R245																		
R244																		
R243			Res.		STIM 0					SYNC								A/D
R242		Port 0	Port 4															
R241	Res.									Res.								
R240																		

Note 1: Depending on device. See device summary on page 1.

Table 4. Detailed Register Map

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
N/A	I/O Port 0:5	R224	P0DR	Port 0 Data Register	FF	68
		R226	P2DR	Port 2 Data Register	FF	
		R227	P3DR	Port 3 Data Register	FF	
		R228	P4DR	Port 4 Data Register	FF	
		R229	P5DR	Port 5 Data Register	FF	
	Core	R230	CICR	Central Interrupt Control Register	87	56
		R231	FLAGR	Flag Register	00	29
		R232	RP0	Pointer 0 Register	xx	31
		R233	RP1	Pointer 1 Register	xx	31
		R234	PPR	Page Pointer Register	xx	33
		R235	MODER	Mode Register	E0	33
		R236	USPHR	User Stack Pointer High Register	xx	35
		R237	USPLR	User Stack Pointer Low Register	xx	35
		R238	SSPHR	System Stack Pointer High Reg.	xx	35
		R239	SSPLR	System Stack Pointer Low Reg.	xx	35
0	INT	R242	EITR	External Interrupt Trigger Register	00	56
		R243	EIPR	External Interrupt Pending Reg.	00	57
		R244	EIMR	External Interrupt Mask-bit Reg.	00	57
		R245	EIPLR	External Interrupt Priority Level Reg.	FF	57
		R246	EIVR	External Interrupt Vector Register	x6	58
		R247	NICR	Nested Interrupt Control	00	58
	WDT	R248	WDTHR	Watchdog Timer High Register	FF	80
		R249	WDTLR	Watchdog Timer Low Register	FF	80
		R250	WDTPR	Watchdog Timer Prescaler Reg.	FF	80
		R251	WDTCR	Watchdog Timer Control Register	12	80
		R252	WCR	Wait Control Register	7F	81
	SPI	R253	SPIDR	SPI Data Register	xx	182
		R254	SPICR	SPI Control Register	00	182
2	I/O Port 0	R240	P0C0	Port 0 Configuration Register 0	00	68
		R241	P0C1	Port 0 Configuration Register 1	00	
		R242	P0C2	Port 0 Configuration Register 2	00	
	I/O Port 2	R248	P2C0	Port 2 Configuration Register 0	00	
		R249	P2C1	Port 2 Configuration Register 1	00	
		R250	P2C2	Port 2 Configuration Register 2	00	
	I/O Port 3	R252	P3C0	Port 3 Configuration Register 0	00	
		R253	P3C1	Port 3 Configuration Register 1	00	
		R254	P3C2	Port 3 Configuration Register 2	00	

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
3	I/O Port 4	R240	P4C0	Port 4 Configuration Register 0	00	68
		R241	P4C1	Port 4 Configuration Register 1	00	
		R242	P4C2	Port 4 Configuration Register 2	00	
	I/O Port 5	R244	P5C0	Port 5 Configuration Register 0	00	
		R245	P5C1	Port 5 Configuration Register 1	00	
		R246	P5C2	Port 5 Configuration Register 2	00	
6	VPS/ WSS	R240	VPSSR	VPS Status Register	00	170
		R241	VPSD0R	VPS Data Register 0	00	170
		R242	VPSD1R	VPS Data Register 1	00	170
		R243	VPSD2R	VPS Data Register 2	00	170
		R244	VPSD3R	VPS Data Register 3	00	171
		R245	VPSD4R	VPS Data Register 4	00	171
		R246	WSSDS0R	WSS Data and Status Register 0	00	171
		R247	WSSDS1R	WSS Data and Status Register 1	00	171
		R248	WSSDS2R	WSS Data and Status Register 2	00	171
		R249	VPSWSSCR	VPS/WSS Control Register	00	172
		R250	WSSDS3R	WSS Data and Status Register 3	00	172
		R251	WSSDS4R	WSS Data and Status Register 4	00	173
		R252	WSSDS5R	WSS Data and Status Register 5	00	173
11	STIM0	R240	ST0HR	Counter High Byte Register	FF	85
		R241	ST0LR	Counter Low Byte Register	FF	85
		R242	ST0PR	Standard Timer Prescaler Register	FF	85
		R243	ST0CR	Standard Timer Control Register	14	85
	STIM1 [1)]	R248	ST1HR	Counter High Byte Register	FF	85
		R249	ST1LR	Counter Low Byte Register	FF	85
		R250	ST1PR	Standard Timer Prescaler Register	FF	85
		R251	ST1CR	Standard Timer Control Register	14	85
21	MMU	R240	DPR0	Data Page Register 0	xx	40
		R241	DPR1	Data Page Register 1	xx	40
		R242	DPR2	Data Page Register 2	xx	40
		R243	DPR3	Data Page Register 3	xx	40
		R244	CSR	Code Segment Register	00	41
		R248	ISR	Interrupt Segment Register	xx	41
		R249	DMASR	DMA Segment Register	xx	41
	Ext.Mem.	R246	EMR2	External Memory Register 2	0F	59

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
32	OSD	R240	HBLANKR	Horizontal Blank Register	03	125
		R241	HPOSR	Horizontal Position Register	03	125
		R242	VPOSR	Vertical Position Register	00	125
		R243	FSCCR	Full Screen Color Control Register	00	126
		R244	HSCR	Header & Status Control Register	2A	127
		R245	NCSR	National Character Set Control Register	00	128
		R246	CHPOSR	Cursor Horizontal Position Register	00	129
		R247	CVPOSR	Cursor Vertical Position Register	00	129
		R248	SCLR	Scrolling Control Low Register	00	130
		R249	SCHR	Scrolling Control High Register	00	131
		R250	DCM0R	Display Control Mode 0 Register	00	133
		R251	DCM1R	Display Control Mode 1 Register	00	134
		R252	TDPR	TDSRAM Pointer Register	00	134
		R253	DE0R	Display Enable 0 Control Register	FF	135
		R254	DE1R	Display Enable 1 Control Register	FF	135
		R255	DE2R	Display Enable 2 Control Register	xF	135
33		R240	DCR	Default Color Register	70	136
		R241	CAPVR	Cursor Absolute Vertical Position Register	00	136
		R246	TDPPR	TDSRAM Page Pointer Register	x0	136
		R247	TDHSPR	TDSRAM Header/Status Pointer Register	x0	136
34	ACQ	R242	ACQAD1R	Acquisition Address Register 1	xx	163
		R243	ACQAD0R	Acquisition Address Register 0	xx	163
		R248	ACQPOR	Acquisition Page Open Register	00	158
		R249	ACQMLR	Acquisition Magazine Locked Register	00	158
		R250	ACQNHRR	Acquisition New Header Received Register	00	158
		R251	ACQPRR	Acquisition Packet Request Register	00	159
		R252	ACQTQMR	Acquisition Teletext Quality Measure Register	00	159
		R253	ACQHD2R	Acquisition Hamming Decoding Register 2	xx	160
		R254	ACQHD1R	Acquisition Hamming Decoding Register 1	xx	160
		R255	ACQHD0R	Acquisition Hamming Decoding Register 0	xx	160
35	SYNC	R242	SCCS0R	Sync Controller Control and Status Register 0	00	144
		R243	SCCS1R	Sync Controller Control and Status Register 1	00	145
	TSU	R248	TXSCR	Teletext Slicer Control Register	06	163
		R249	TXSLIR	Teletext Slicer Initialization Register	0B	164
		R255	PASR	Pre-Amplifier and ADC Selection Register	00	164

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
36	TDSRAM	R240 .. R255	BUF0 .. BUF15	40-byte buffer	xx .. xx	91
37		R240 .. R255	BUF16 .. BUF31		xx .. xx	
38		R240 .. R247	BUF32 .. BUF39		xx .. xx	
		R248	BUFC	TDSRAM Buffer Control Register	08	92
		R250	MTBSA1	Multi-byte Transfer Start Address Register 1	80	91
		R251	MTBSA0	Multi-byte Transfer Start Address Register 0	00	91
		R252	CONFIG	TDSRAM Interface Configuration Register	06	93
39	TCC	R251	PXCCR	PLL Clock Control Register	00	68
		R252	SLCCR	Slicer Clock Control Register	00	68
		R253	MCCR	Main Clock Control Register	00	67
		R254	SKCCR	Skew Clock Control Register	00	67
44	I²C [1)]	R240	I²COAR	Own Address Register	00	on page 189189
		R241	I²CFQR	Frequency Register	00	190
		R242	I²CCTR	Control Register	01	191
		R243	I²CDR	Data Register	00	192
		R244	I²CSTR2	Status Register 2	00	192
		R245	I²CSTR1	Status Register 1	00	193
55	RCCU	R251	PCONF	PLL Configuration Register	07	63
		R254	SDRATH	Clock Slow Down Unit Ratio Register	2x,4x or 00	63

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
59	PWM	R240	CM0	Compare Register 0	00	207
		R241	CM1	Compare Register 1	00	207
		R242	CM2	Compare Register 2	00	207
		R243	CM3	Compare Register 3	00	207
		R244	CM4	Compare Register 4	00	207
		R245	CM5	Compare Register 5	00	207
		R246	CM6	Compare Register 6	00	207
		R247	CM7	Compare Register 7	00	207
		R248	ACR	Autoclear Register	FF	208
		R249	CCR	Counter Register	00	208
		R250	PCTL	Prescaler and Control Register	0C	208
		R251	OCPL	Output Complement Register	00	209
		R252	OER	Output Enable Register	00	209
	VS	R254	VSDR1	Data and Control Register 1	00	204
		R255	VSDR2	Data Register 2	00	204
62	ADC	R240	ADDTR	Channel i Data Register	xx	199
		R241	ADCLR	Control Logic Register	00	198
		R242	ADINT	AD Interrupt Register	01	199

Note: xx denotes a byte with an undefined value, however some of the bits may have defined values. Refer to register description for details.

Note 1: Depending on device. See device summary on page 1.

2 DEVICE ARCHITECTURE

2.1 CORE ARCHITECTURE

The ST9 Core or Central Processing Unit (CPU) features a highly optimised instruction set, capable of handling bit, byte (8-bit) and word (16-bit) data, as well as BCD and Boolean formats; 14 addressing modes are available.

Four independent buses are controlled by the Core: a 16-bit Memory bus, an 8-bit Register data bus, an 8-bit Register address bus and a 6-bit Interrupt/DMA bus which connects the interrupt and DMA controllers in the on-chip peripherals with the Core.

This multiple bus architecture affords a high degree of pipelining and parallel operation, thus making the ST9 family devices highly efficient, both for numerical calculation, data handling and with regard to communication with on-chip peripheral resources.

2.2 MEMORY SPACES

There are two separate memory spaces:

– The Register File, which comprises 240 8-bit registers, arranged as 15 groups (Group 0 to E), each containing sixteen 8-bit registers plus up to 64 pages of 16 registers mapped in Group F, which hold data and control bits for the on-chip peripherals and I/Os.

– A single linear memory space accommodating both program and data. All of the physically separate memory areas, including the internal ROM, internal RAM and external memory are mapped in this common address space. The total addressable memory space of 4 Mbytes (limited by the size of on-chip memory and the number of external address pins) is arranged as 64 segments of 64 Kbytes. Each segment is further subdivided into four pages of 16 Kbytes, as illustrated in Figure 1. A Memory Management Unit uses a set of pointer registers to address a 22-bit memory field using 16-bit address-based instructions.

2.2.1 Register File

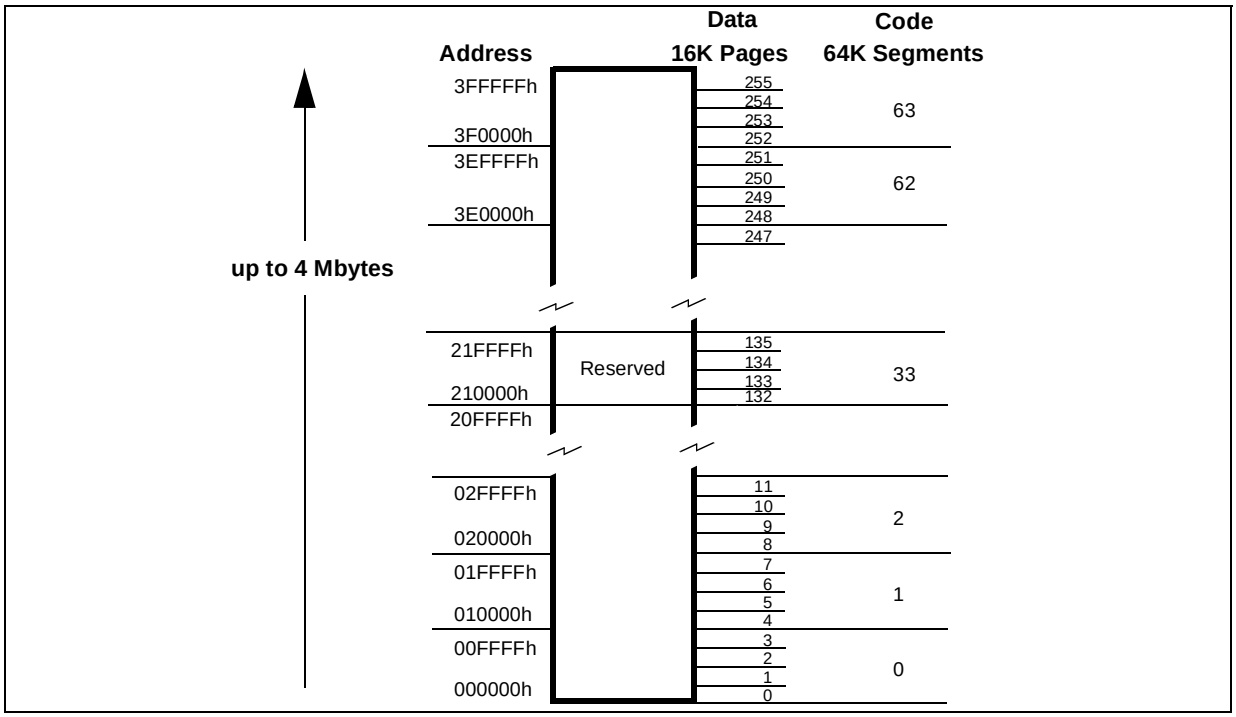
The Register File consists of (see Figure 2):

– 224 general purpose registers (Group 0 to D, registers R0 to R223)

– 6 system registers in the System Group (Group E, registers R224 to R239)

– Up to 64 pages, depending on device configuration, each containing up to 16 registers, mapped to Group F (R240 to R255), see Figure 3.

Figure 7. Single Program and Data Memory Address Space



MEMORY SPACES (Cont'd)

Figure 8. Register Groups

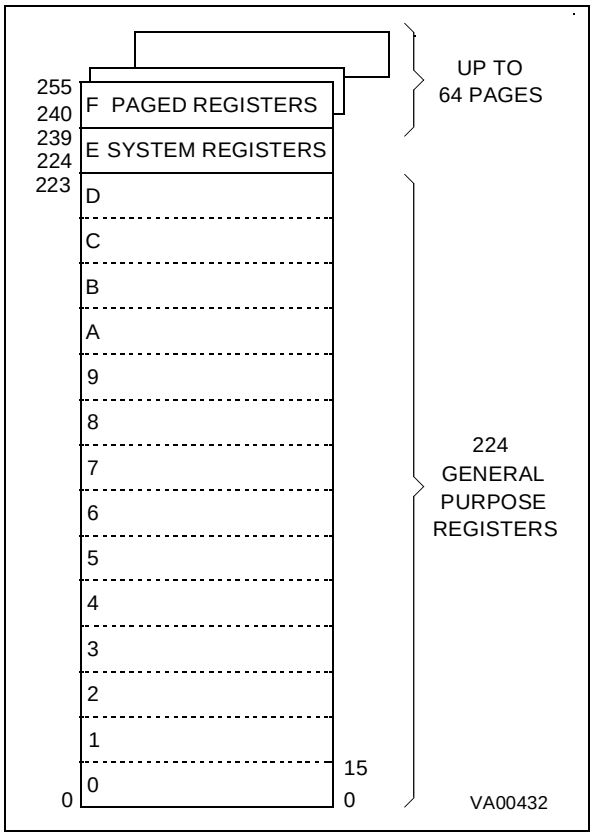


Figure 9. Page Pointer for Group F mapping

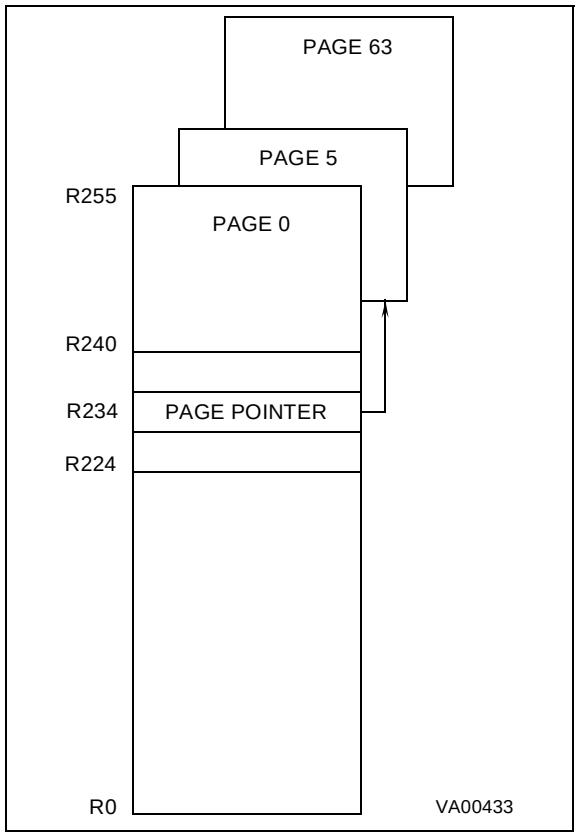
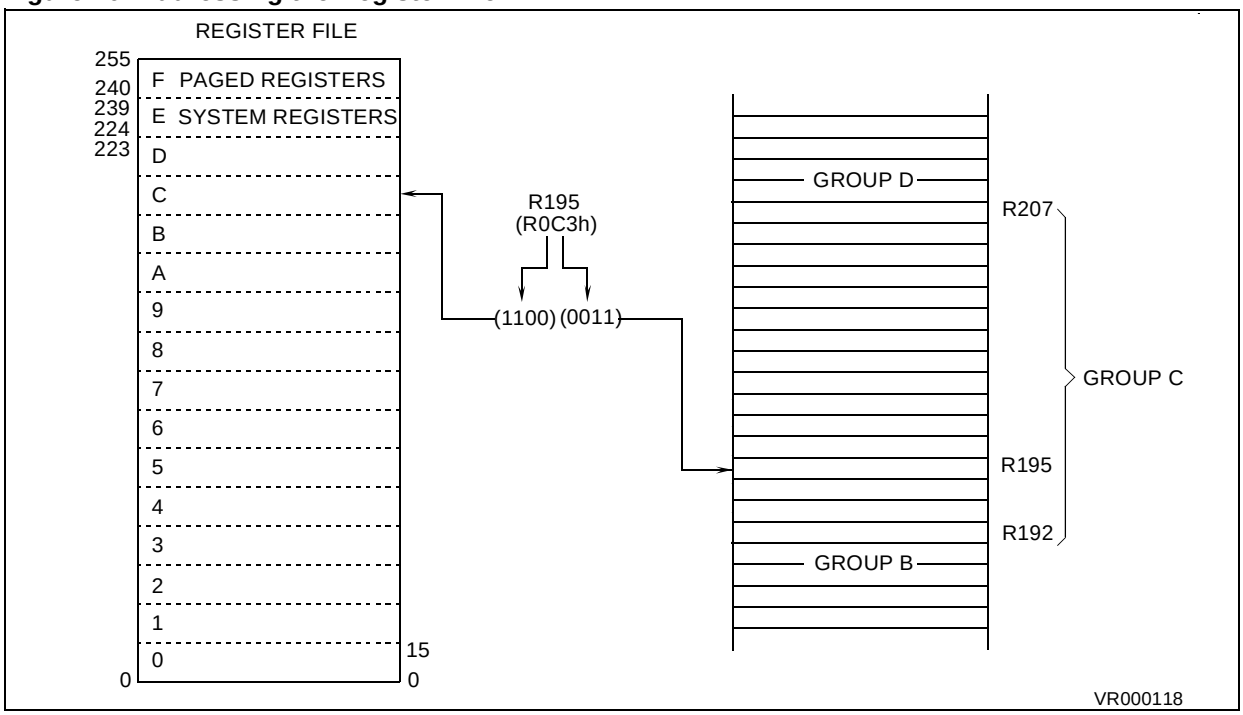


Figure 10. Addressing the Register File



MEMORY SPACES (Cont'd)

2.2.2 Register Addressing

Register File registers, including Group F paged registers (but excluding Group D), may be addressed explicitly by means of a decimal, hexadecimal or binary address; thus R231, RE7h and R11100111b represent the same register (see Figure 4). Group D registers can only be addressed in Working Register mode.

Note that an upper case "R" is used to denote this direct addressing mode.

Working Registers

Certain types of instruction require that registers be specified in the form "rx", where x is in the range 0 to 15: these are known as Working Registers.

Note that a lower case "r" is used to denote this indirect addressing mode.

Two addressing schemes are available: a single group of 16 working registers, or two separately mapped groups, each consisting of 8 working registers. These groups may be mapped starting at any 8 or 16 byte boundary in the register file by means of dedicated pointer registers. This technique is described in more detail in Section 1.3.3, and illustrated in Figure 5 and in Figure 6.

System Registers

The 16 registers in Group E (R224 to R239) are System registers and may be addressed using any of the register addressing modes. These registers are described in greater detail in Section 1.3.

Paged Registers

Up to 64 pages, each containing 16 registers, may be mapped to Group F. These are addressed using any register addressing mode, in conjunction with the Page Pointer register, R234, which is one of the System registers. This register selects the page to be mapped to Group F and, once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Therefore if the Page Pointer, R234, is set to 5, the instructions:

```
spp #5
ld R242, r4
```

will load the contents of working register r4 into the third register of page 5 (R242).

These paged registers hold data and control information relating to the on-chip peripherals, each peripheral always being associated with the same pages and registers to ensure code compatibility between ST9 devices. The number of these registers therefore depends on the peripherals which are present in the specific ST9 family device. In other words, pages only exist if the relevant peripheral is present.

Table 5. Register File Organization

Hex. Address	Decimal Address	Function	Register File Group
F0-FF	240-255	Paged Registers	Group F
E0-EF	224-239	System Registers	Group E
D0-DF	208-223	General Purpose Registers	Group D
C0-CF	192-207		Group C
B0-BF	176-191		Group B
A0-AF	160-175		Group A
90-9F	144-159		Group 9
80-8F	128-143		Group 8
70-7F	112-127		Group 7
60-6F	96-111		Group 6
50-5F	80-95		Group 5
40-4F	64-79		Group 4
30-3F	48-63		Group 3
20-2F	32-47		Group 2
10-1F	16-31		Group 1
00-0F	00-15		Group 0

2.3 SYSTEM REGISTERS

The System registers are listed in Table 2 System Registers (Group E). They are used to perform all the important system settings. Their purpose is described in the following pages. Refer to the chapter dealing with I/O for a description of the PORT[5:0] Data registers.

Table 6. System Registers (Group E)

R239 (EFh)	SSPLR
R238 (EEh)	SSPHR
R237 (EDh)	USPLR
R236 (ECh)	USPHR
R235 (EBh)	MODE REGISTER
R234 (EAh)	PAGE POINTER REGISTER
R233 (E9h)	REGISTER POINTER 1
R232 (E8h)	REGISTER POINTER 0
R231 (E7h)	FLAG REGISTER
R230 (E6h)	CENTRAL INT. CNTL REG
R229 (E5h)	PORT5 DATA REG.
R228 (E4h)	PORT4 DATA REG.
R227 (E3h)	PORT3 DATA REG.
R226 (E2h)	PORT2 DATA REG.
R225 (E1h)	PORT1 DATA REG.
R224 (E0h)	PORT0 DATA REG.

2.3.1 Central Interrupt Control Register

Please refer to the "INTERRUPT" chapter for a detailed description of the ST9 interrupt philosophy.

CENTRAL INTERRUPT CONTROL REGISTER (CICR)
R230 - Read/Write
Register Group: E (System)
Reset Value: 1000 0111 (87h)

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

Bit 7 = **GCEN**: *Global Counter Enable*.
This bit is the Global Counter Enable of the Multifunction Timers. The GCEN bit is ANDed with the CE bit in the TCR Register (only in devices featuring the MFT Multifunction Timer) in order to enable the Timers when both bits are set. This bit is set after the Reset cycle.

Note: If an MFT is not included in the ST9 device, then this bit has no effect.

Bit 6 = **TLIP**: *Top Level Interrupt Pending*.
This bit is set by hardware when a Top Level Interrupt Request is recognized. This bit can also be set by software to simulate a Top Level Interrupt Request.
0: No Top Level Interrupt pending
1: Top Level Interrupt pending

Bit 5 = **TLI**: *Top Level Interrupt bit*.
0: Top Level Interrupt is acknowledged depending on the TLNM bit in the NICR Register.
1: Top Level Interrupt is acknowledged depending on the IEN and TLNM bits in the NICR Register (described in the Interrupt chapter).

Bit 4 = **IEN**: *Interrupt Enable .*
This bit is cleared by interrupt acknowledgement, and set by interrupt return (`iret`). IEN is modified implicitly by `iret`, `ei` and `di` instructions or by an interrupt acknowledge cycle. It can also be explicitly written by the user, but only when no interrupt is pending. Therefore, the user should execute a `di` instruction (or guarantee by other means that no interrupt request can arrive) before any write operation to the CICR register.
0: Disable all interrupts except Top Level Interrupt.
1: Enable Interrupts

Bit 3 = **IAM**: *Interrupt Arbitration Mode*.
This bit is set and cleared by software to select the arbitration mode.
0: Concurrent Mode
1: Nested Mode.

Bits 2:0 = **CPL[2:0]**: *Current Priority Level*.
These three bits record the priority level of the routine currently running (i.e. the Current Priority Level, CPL). The highest priority level is represented by 000, and the lowest by 111. The CPL bits can be set by hardware or software and provide the reference according to which subsequent interrupts are either left pending or are allowed to interrupt the current interrupt service routine. When the current interrupt is replaced by one of a higher priority, the current priority value is automatically stored until required in the NICR register.

SYSTEM REGISTERS (Cont'd)

2.3.2 Flag Register

The Flag Register contains 8 flags which indicate the CPU status. During an interrupt, the flag register is automatically stored in the system stack area and recalled at the end of the interrupt service routine, thus returning the CPU to its original status.

This occurs for all interrupts and, when operating in nested mode, up to seven versions of the flag register may be stored.

FLAG REGISTER (FLAGR)
R231- Read/Write
Register Group: E (System)
Reset value: 0000 0000 (00h**)**

7							0
C	Z	S	V	DA	H	-	DP

Bit 7 = **C**: *Carry Flag*.
The carry flag is affected by:
Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Shift Left Arithmetic (`sla, slaw`),
Swap Nibbles (`swap`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`).

When set, it generally indicates a carry out of the most significant bit position of the register being used as an accumulator (bit 7 for byte operations and bit 15 for word operations).

The carry flag can be set by the Set Carry Flag (`scf`) instruction, cleared by the Reset Carry Flag (`rcf`) instruction, and complemented by the Complement Carry Flag (`ccf`) instruction.

Bit 6 = **Z:** *Zero Flag*. The Zero flag is affected by:
Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Shift Left Arithmetic (`sla, slaw`),
Swap Nibbles (`swap`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),
Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the contents of the register being used as an accumulator become zero, following one of the above operations.

Bit 5 = **S**: *Sign Flag*.
The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for a byte operation) or bit 15 (for a word operation) of the register used as an accumulator is one.

Bit 4 = **V**: *Overflow Flag*.
The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in two's-complement notation.

Bit 3 = **DA**: *Decimal Adjust Flag*.
The DA flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (`da`) operation can perform its function correctly. The DA flag cannot normally be used as a test condition by the programmer.

Bit 2 = **H**: *Half Carry Flag.*
The H flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The H flag is used by the Decimal Adjust (`da`) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result. Like the DA flag, this flag is not normally accessed by the user.

Bit 1 = Reserved bit (must be 0).

Bit 0 = **DP**: *Data/Program Memory Flag*.
This bit indicates the memory area addressed. Its value is affected by the Set Data Memory (`sdm`) and Set Program Memory (`spm`) instructions. Refer to the Memory Management Unit for further details.

SYSTEM REGISTERS (Cont'd)

If the bit is set, data is accessed using the Data Pointers (DPRs registers), otherwise it is pointed to by the Code Pointer (CSR register); therefore, the user initialization routine must include a Sdm instruction. Note that code is always pointed to by the Code Pointer (CSR).

Note: In the current ST9 devices, the DP flag is only for compatibility with software developed for the first generation of ST9 devices. With the single memory addressing space, its use is now redundant. It must be kept to 1 with a Sdm instruction at the beginning of the program to ensure a normal use of the different memory pointers.

2.3.3 Register Pointing Techniques

Two registers within the System register group, are used as pointers to the working registers. Register Pointer 0 (R232) may be used on its own as a single pointer to a 16-register working space, or in conjunction with Register Pointer 1 (R233), to point to two separate 8-register spaces.

For the purpose of register pointing, the 16 register groups of the register file are subdivided into 32 8-register blocks. The values specified with the Set Register Pointer instructions refer to the blocks to be pointed to in twin 8-register mode, or to the lower 8-register block location in single 16-register mode.

The Set Register Pointer instructions srp, srp0 and srp1 automatically inform the CPU whether the Register File is to operate in single 16-register mode or in twin 8-register mode. The srp instruction selects the single 16-register group mode and specifies the location of the lower 8-register block, while the srp0 and srp1 instructions automatically select the twin 8-register group mode and specify the locations of each 8-register block.

There is no limitation on the order or position of these register groups, other than that they must start on an 8-register boundary in twin 8-register mode, or on a 16-register boundary in single 16-register mode.

The block number should always be an even number in single 16-register mode. The 16-register group will always start at the block whose number is the nearest even number equal to or lower than the block number specified in the srp instruction. Avoid using odd block numbers, since this can be confusing if twin mode is subsequently selected.

Thus:

srp #3 will be interpreted as srp #2 and will allow using R16 ..R31 as r0 .. r15.

In single 16-register mode, the working registers are referred to as r0 to r15. In twin 8-register mode, registers r0 to r7 are in the block pointed to by RP0 (by means of the srp0 instruction), while registers r8 to r15 are in the block pointed to by RP1 (by means of the srp1 instruction).

Caution: *Group D registers can only be accessed as working registers using the Register Pointers, or by means of the Stack Pointers. They cannot be addressed explicitly in the form "Rxxx".*

SYSTEM REGISTERS (Cont'd)

POINTER 0 REGISTER (RP0)
R232 - Read/Write
Register Group: E (System)
Reset Value: xxxx xx00 (xxh)

7							0
RG4	RG3	RG2	RG1	RG0	RPS	0	0

Bits 7:3 = **RG[4:0]**: *Register Group number.*
These bits contain the number (in the range 0 to 31) of the register block specified in the `srp0` or `srp` instructions. In single 16-register mode the number indicates the lower of the two 8-register blocks to which the 16 working registers are to be mapped, while in twin 8-register mode it indicates the 8-register block to which `r0` to `r7` are to be mapped.

Bit 2 = **RPS**: *Register Pointer Selector.*
This bit is set by the instructions `srp0` and `srp1` to indicate that the twin register pointing mode is selected. The bit is reset by the `srp` instruction to indicate that the single register pointing mode is selected.
0: Single register pointing mode
1: Twin register pointing mode

Bits 1:0: Reserved. Forced by hardware to zero.

POINTER 1 REGISTER (RP1)
R233 - Read/Write
Register Group: E (System)
Reset Value: xxxx xx00 (xxh)

7							0
RG4	RG3	RG2	RG1	RG0	RPS	0	0

This register is only used in the twin register pointing mode. When using the single register pointing mode, or when using only one of the twin register groups, the RP1 register must be considered as RESERVED and may NOT be used as a general purpose register.

Bits 7:3 = **RG[4:0]:** *Register Group number.*
These bits contain the number (in the range 0 to 31) of the 8-register block specified in the `srp1` instruction, to which `r8` to `r15` are to be mapped.

Bit 2 = **RPS**: *Register Pointer Selector.*
This bit is set by the `srp0` and `srp1` instructions to indicate that the twin register pointing mode is selected. The bit is reset by the `srp` instruction to indicate that the single register pointing mode is selected.
0: Single register pointing mode
1: Twin register pointing mode

Bits 1:0: Reserved. Forced by hardware to zero.

SYSTEM REGISTERS (Cont'd)

Figure 11. Pointing to a single group of 16 registers

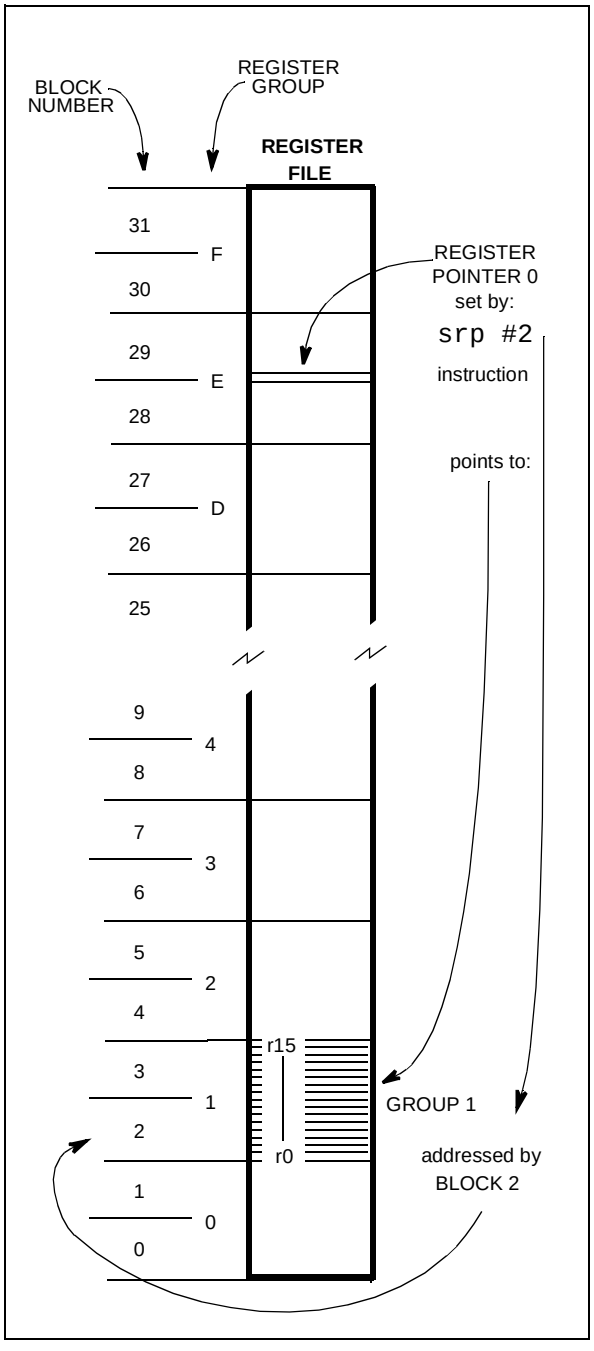
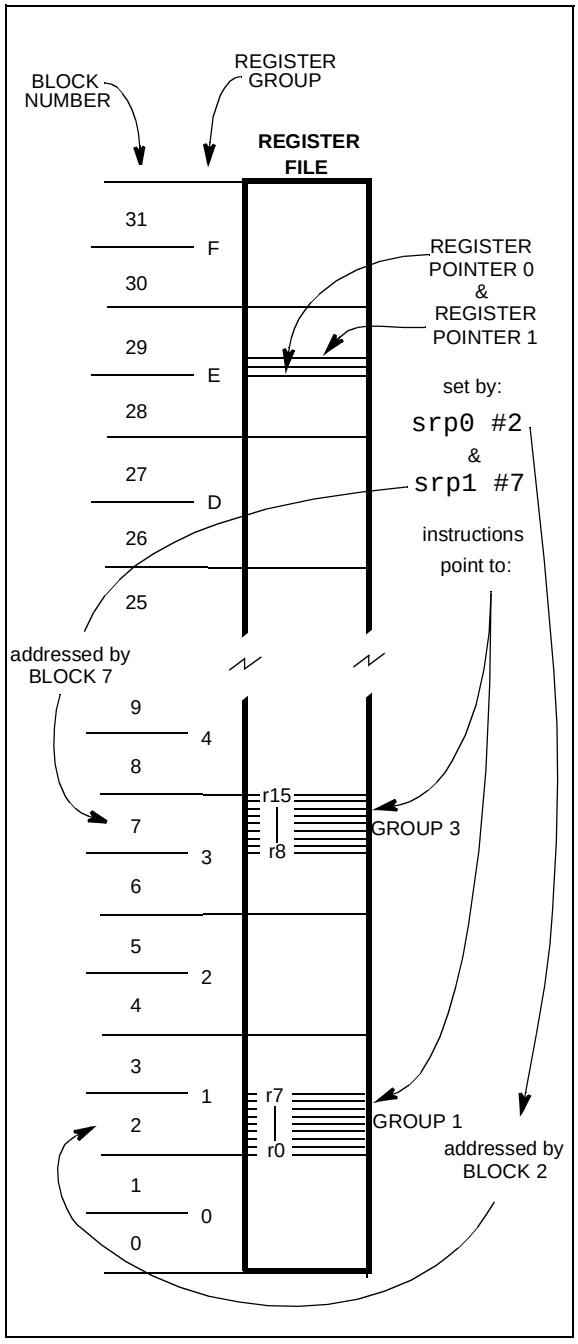


Figure 12. Pointing to two groups of 8 registers



SYSTEM REGISTERS (Cont'd)

2.3.4 Paged Registers

Up to 64 pages, each containing 16 registers, may be mapped to Group F. These paged registers hold data and control information relating to the on-chip peripherals, each peripheral always being associated with the same pages and registers to ensure code compatibility between ST9 devices. The number of these registers depends on the peripherals present in the specific ST9 device. In other words, pages only exist if the relevant peripheral is present.

The paged registers are addressed using the normal register addressing modes, in conjunction with the Page Pointer register, R234, which is one of the System registers. This register selects the page to be mapped to Group F and, once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Thus the instructions:

```
spp #5
ld R242, r4
```

will load the contents of working register r4 into the third register of page 5 (R242).

Warning: During an interrupt, the PPR register is not saved automatically in the stack. If needed, it should be saved/restored by the user within the interrupt routine.

PAGE POINTER REGISTER (PPR)
R234 - Read/Write
Register Group: E (System)
Reset value: xxxx xx00 (xxh**)**

7							0
PP5	PP4	PP3	PP2	PP1	PP0	0	0

Bits 7:2 = **PP[5:0]**: *Page Pointer*.
These bits contain the number (in the range 0 to 63) of the page specified in the spp instruction. Once the page pointer has been set, there is no need to refresh it unless a different page is required.

Bits 1:0: Reserved. Forced by hardware to 0.

2.3.5 Mode Register

The Mode Register allows control of the following operating parameters:

– Selection of internal or external System and User Stack areas,
– Management of the clock frequency,
– Enabling of Bus request and Wait signals when interfacing to external memory.

MODE REGISTER (MODER)
R235 - Read/Write
Register Group: E (System)
Reset value: 1110 0000 (E0h)

7							0
SSP	USP	DIV2	PRS2	PRS1	PRS0	BRQEN	HIMP

Bit 7 = **SSP**: *System Stack Pointer*.
This bit selects an internal or external System Stack area.
0: External system stack area, in memory space.
1: Internal system stack area, in the Register File (reset state).

Bit 6 = **USP**: *User Stack Pointer*.
This bit selects an internal or external User Stack area.
0: External user stack area, in memory space.
1: Internal user stack area, in the Register File (reset state).

Bit 5 = **DIV2**: *Crystal Oscillator Clock Divided by 2*.
This bit controls the divide-by-2 circuit operating on the crystal oscillator clock (CLOCK1).
0: Clock divided by 1
1: Clock divided by 2

Bits 4:2 = **PRS[2:0]**: *CPUCLK Prescaler*.
These bits load the prescaler division factor for the internal clock (INTCLK). The prescaler factor selects the internal clock frequency, which can be divided by a factor from 1 to 8. Refer to the Reset and Clock Control chapter for further information.

Bit 1 = **BRQEN**: *Bus Request Enable*.
0: External Memory Bus Request disabled
1: External Memory Bus Request enabled on $\overline{\text{BREQ}}$ pin (where available).

Note: Disregard this bit if $\overline{\text{BREQ}}$ pin is not available.

Bit 0 = **HIMP**: *High Impedance Enable*.
When any of Ports 0, 1, 2 or 6 depending on device configuration, are programmed as Address and Data lines to interface external Memory, these lines and the Memory interface control lines (AS, DS, R/W) can be forced into the High Impedance

SYSTEM REGISTERS (Cont'd)

state by setting the HIMP bit. When this bit is reset, it has no effect.

Setting the HIMP bit is recommended for noise reduction when only internal Memory is used.

If Port 1 and/or 2 are declared as an address AND as an I/O port (for example: P10... P14 = Address, and P15... P17 = I/O), the HIMP bit has no effect on the I/O lines.

2.3.6 Stack Pointers

Two separate, double-register stack pointers are available: the System Stack Pointer and the User Stack Pointer, both of which can address registers or memory.

The stack pointers point to the "bottom" of the stacks which are filled using the push commands and emptied using the pop commands. The stack pointer is automatically pre-decremented when data is "pushed" in and post-incremented when data is "popped" out.

The push and pop commands used to manage the System Stack may be addressed to the User Stack by adding the suffix "u". To use a stack instruction for a word, the suffix "w" is added. These suffixes may be combined.

When bytes (or words) are "popped" out from a stack, the contents of the stack locations are unchanged until fresh data is loaded. Thus, when data is "popped" from a stack area, the stack contents remain unchanged.

Note: Instructions such as: `pushuw RR236` or `pushw RR238`, as well as the corresponding `pop` instructions (where R236 & R237, and R238 & R239 are themselves the user and system stack pointers respectively), must not be used, since the pointer values are themselves automatically changed by the `push` or `pop` instruction, thus corrupting their value.

System Stack

The System Stack is used for the temporary storage of system and/or control data, such as the Flag register and the Program counter.

The following automatically push data onto the System Stack:

– **Interrupts**

When entering an interrupt, the PC and the Flag Register are pushed onto the System Stack. If the ENCSR bit in the EMR2 register is set, then the Code Segment Register is also pushed onto the System Stack.

– **Subroutine Calls**

When a `call` instruction is executed, only the PC is pushed onto stack, whereas when a `calls` instruction (call segment) is executed, both the PC and the Code Segment Register are pushed onto the System Stack.

– **Link Instruction**

The `link` or `linku` instructions create a C language stack frame of user-defined length in the System or User Stack.

All of the above conditions are associated with their counterparts, such as return instructions, which pop the stored data items off the stack.

User Stack

The User Stack provides a totally user-controlled stacking area.

The User Stack Pointer consists of two registers, R236 and R237, which are both used for addressing a stack in memory. When stacking in the Register File, the User Stack Pointer High Register, R236, becomes redundant but must be considered as reserved.

Stack Pointers

Both System and User stacks are pointed to by double-byte stack pointers. Stacks may be set up in RAM or in the Register File. Only the lower byte will be required if the stack is in the Register File. The upper byte must then be considered as reserved and must not be used as a general purpose register.

The stack pointer registers are located in the System Group of the Register File, this is illustrated in Table 2 System Registers (Group E).

Stack Location

Care is necessary when managing stacks as there is no limit to stack sizes apart from the bottom of any address space in which the stack is placed. Consequently programmers are advised to use a stack pointer value as high as possible, particularly when using the Register File as a stacking area.

Group D is a good location for a stack in the Register File, since it is the highest available area. The stacks may be located anywhere in the first 14 groups of the Register File (internal stacks) or in RAM (external stacks).

Note. Stacks must not be located in the Paged Register Group or in the System Register Group.

SYSTEM REGISTERS (Cont'd)

USER STACK POINTER HIGH REGISTER (USPHR)
R236 - Read/Write
Register Group: E (System)
Reset value: undefined

7							0
USP15	USP14	USP13	USP12	USP11	USP10	USP9	USP8

USER STACK POINTER LOW REGISTER (USPLR)
R237 - Read/Write
Register Group: E (System)
Reset value: undefined

7							0
USP7	USP6	USP5	USP4	USP3	USP2	USP1	USP0

SYSTEM STACK POINTER HIGH REGISTER (SSPHR)
R238 - Read/Write
Register Group: E (System)
Reset value: undefined

7							0
SSP15	SSP14	SSP13	SSP12	SSP11	SSP10	SSP9	SSP8

SYSTEM STACK POINTER LOW REGISTER (SSPLR)
R239 - Read/Write
Register Group: E (System)
Reset value: undefined

7							0
SSP7	SSP6	SSP5	SSP4	SSP3	SSP2	SSP1	SSP0

Figure 13. Internal Stack Mode

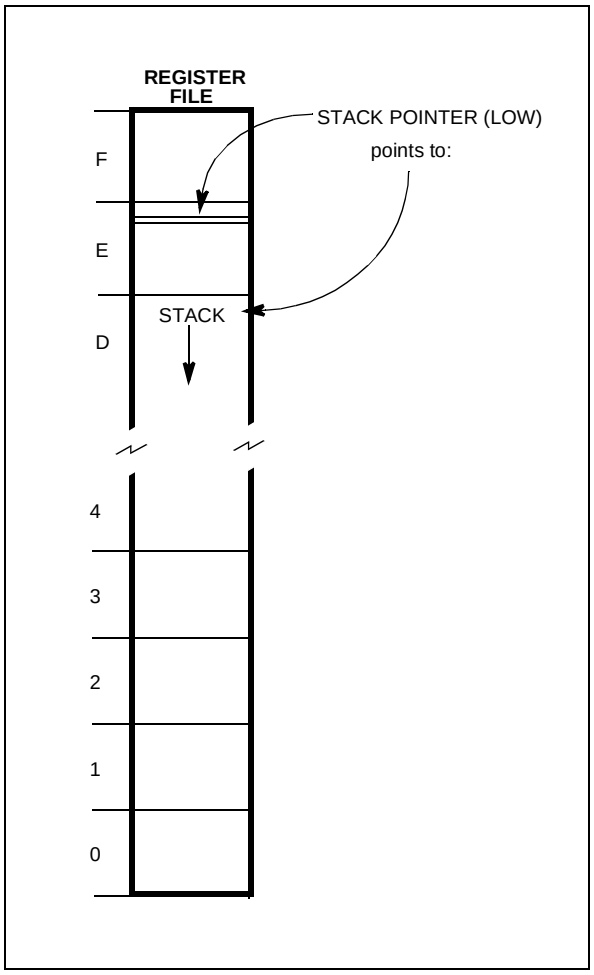
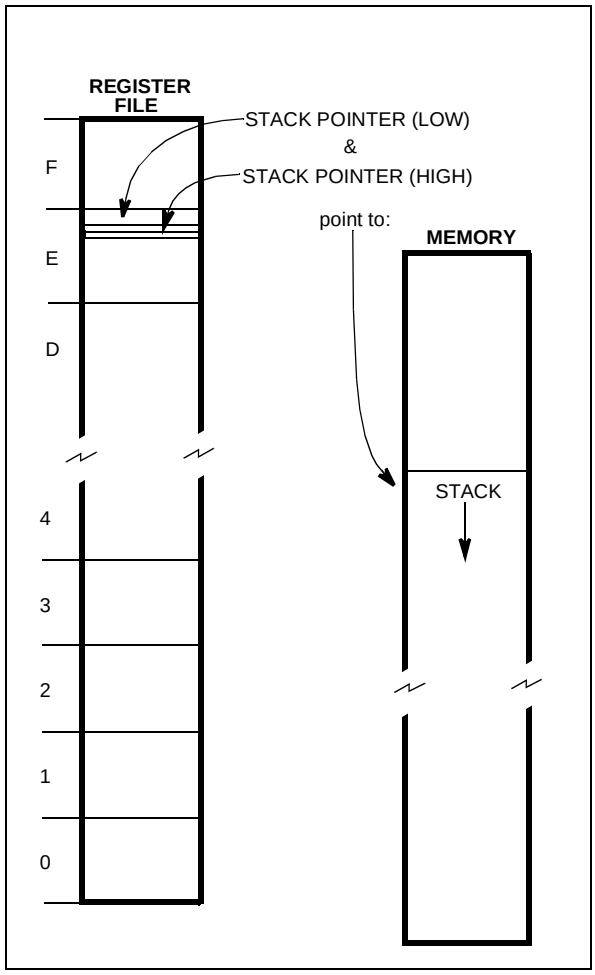


Figure 14. External Stack Mode



2.4 MEMORY ORGANIZATION

Code and data are accessed within the same linear address space. All of the physically separate memory areas, including the internal ROM, internal RAM and external memory are mapped in a common address space.

The ST9 provides a total addressable memory space of 4 Mbytes. This address space is arranged as 64 segments of 64 Kbytes; each segment is again subdivided into four 16 Kbyte pages.

The mapping of the various memory areas (internal RAM or ROM, external memory) differs from device to device. Each 64-Kbyte physical memory segment is mapped either internally or externally; if the memory is internal and smaller than 64 Kbytes, the remaining locations in the 64-Kbyte segment are not used (reserved).

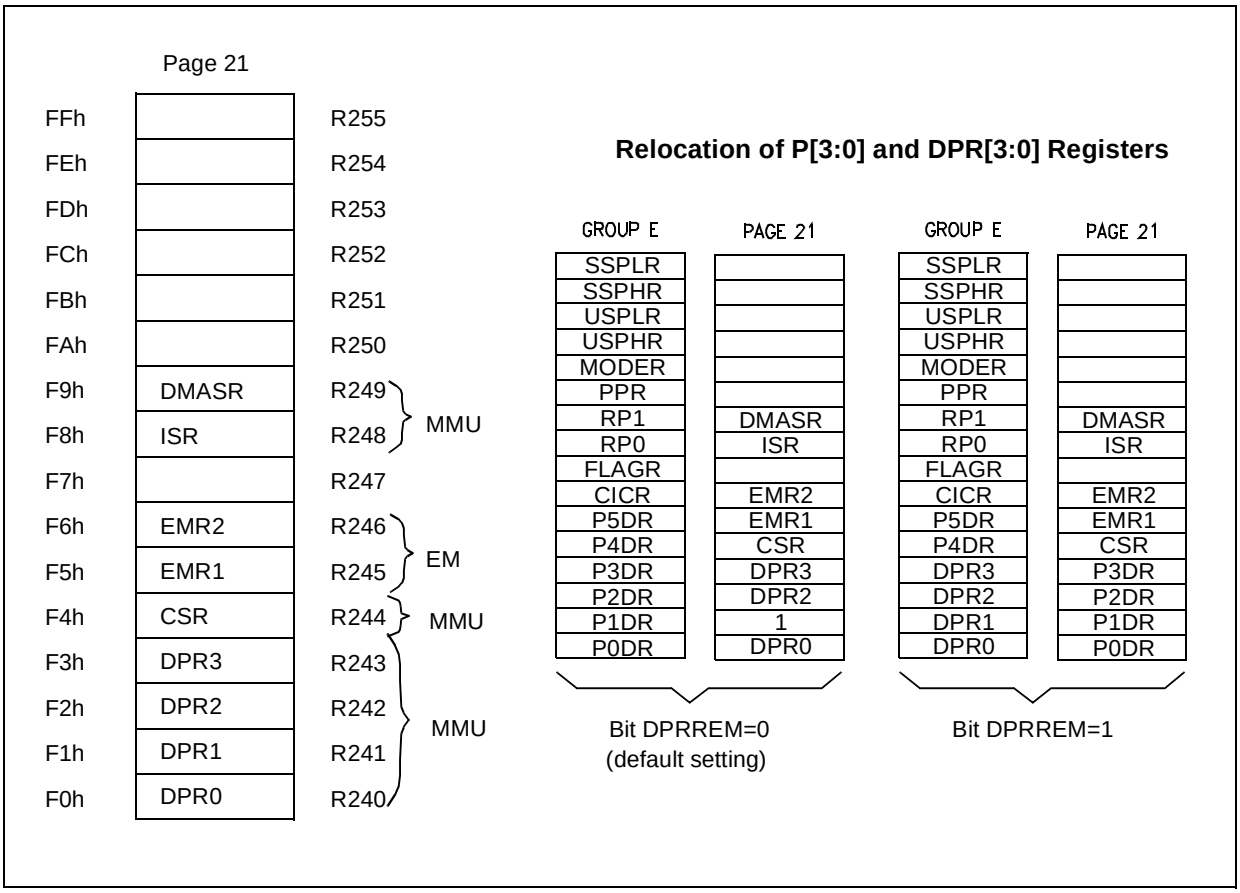
Refer to the Register and Memory Map Chapter for more details on the memory map.

2.5 MEMORY MANAGEMENT UNIT

The CPU Core includes a Memory Management Unit (MMU) which must be programmed to perform memory accesses (even if external memory is not used).

The MMU is controlled by 7 registers and 2 bits (ENCSR and DPRREM) present in EMR2, which may be written and read by the user program. These registers are mapped within group F, Page 21 of the Register File. The 7 registers may be sub-divided into 2 main groups: a first group of four 8-bit registers (DPR[3:0]), and a second group of three 6-bit registers (CSR, ISR, and DMASR). The first group is used to extend the address during Data Memory access (DPR[3:0]). The second is used to manage Program and Data Memory accesses during Code execution (CSR), Interrupts Service Routines (ISR or CSR), and DMA transfers (DMASR or ISR).

Figure 15. Page 21 Registers



2.6 ADDRESS SPACE EXTENSION

To manage 4 Mbytes of addressing space, it is necessary to have 22 address bits. The MMU adds 6 bits to the usual 16-bit address, thus translating a 16-bit virtual address into a 22-bit physical address. There are 2 different ways to do this depending on the memory involved and on the operation being performed.

2.6.1 Addressing 16-Kbyte Pages

This extension mode is implicitly used to address Data memory space if no DMA is being performed.

The Data memory space is divided into 4 pages of 16 Kbytes. Each one of the four 8-bit registers (DPR[3:0], Data Page Registers) selects a different 16-Kbyte page. The DPR registers allow access to the entire memory space which contains 256 pages of 16 Kbytes.

Data paging is performed by extending the 14 LSB of the 16-bit address with the contents of a DPR register. The two MSBs of the 16-bit address are interpreted as the identification number of the DPR register to be used. Therefore, the DPR registers are involved in the following virtual address ranges:

DPR0: from 0000h to 3FFFh;

DPR1: from 4000h to 7FFFh;

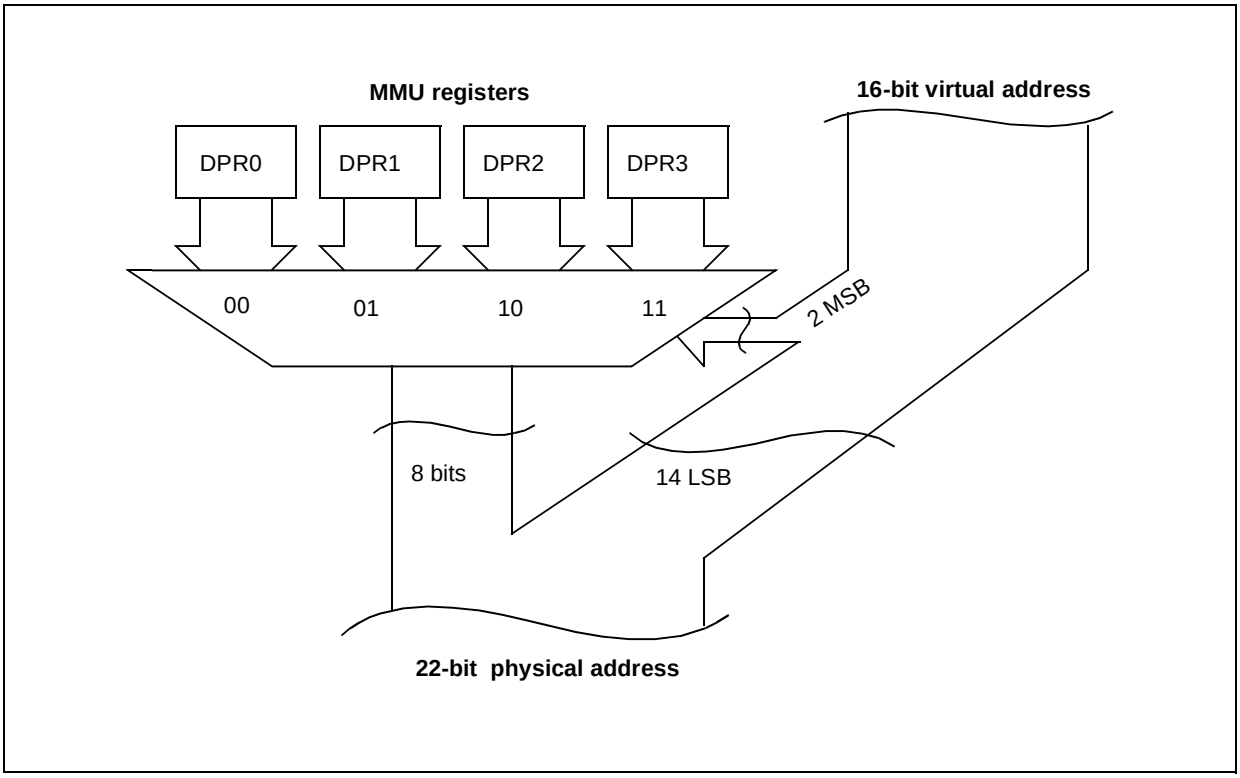
DPR2: from 8000h to BFFFh;

DPR3: from C000h to FFFFh.

The contents of the selected DPR register specify one of the 256 possible data memory pages. This 8-bit data page number, in addition to the remaining 14-bit page offset address forms the physical 22-bit address (see Figure 10).

A DPR register cannot be modified via an addressing mode that uses the same DPR register. For instance, the instruction “POPW DPR0” is legal only if the stack is kept either in the register file or in a memory location above 8000h, where DPR2 and DPR3 are used. Otherwise, since DPR0 and DPR1 are modified by the instruction, unpredictable behaviour could result.

Figure 16. Addressing via DPR[3:0]



ADDRESS SPACE EXTENSION (Cont'd)

2.6.2 Addressing 64-Kbyte Segments

This extension mode is used to address Data memory space during a DMA and Program memory space during any code execution (normal code and interrupt routines).

Three registers are used: CSR, ISR, and DMASR. The 6-bit contents of one of the registers CSR, ISR, or DMASR define one out of 64 Memory segments of 64 Kbytes within the 4 Mbytes address space. The register contents represent the 6 MSBs of the memory address, whereas the 16 LSBs of the address (intra-segment address) are given by the virtual 16-bit address (see Figure 11).

2.7 MMU REGISTERS

The MMU uses 7 registers mapped into Group F, Page 21 of the Register File and 2 bits of the EMR2 register.

Most of these registers do not have a default value after reset.

2.7.1 DPR[3:0]: Data Page Registers

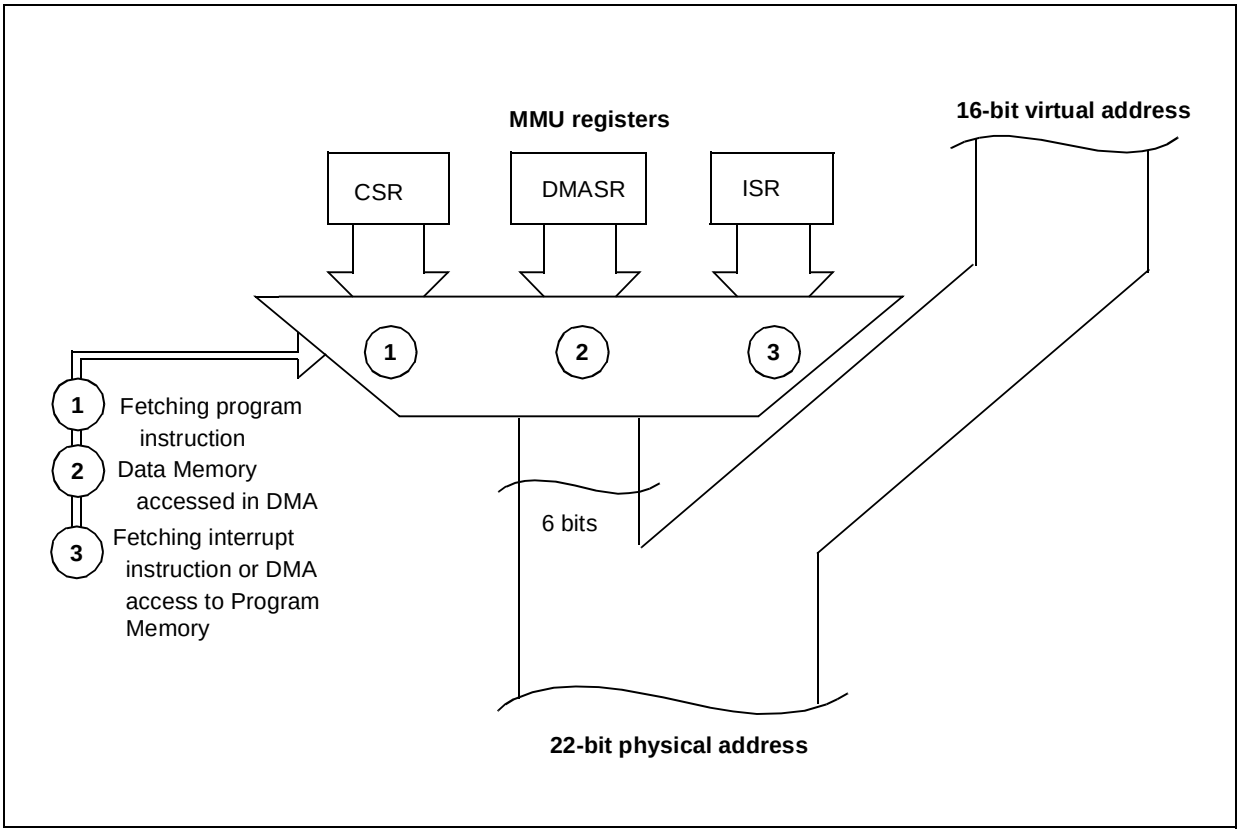
The DPR[3:0] registers allow access to the entire 4 Mbyte memory space composed of 256 pages of 16 Kbytes.

2.7.1.1 Data Page Register Relocation

If these registers are to be used frequently, they may be relocated in register group E, by programming bit 5 of the EMR2-R246 register in page 21. If this bit is set, the DPR[3:0] registers are located at R224-227 in place of the Port 0-3 Data Registers, which are re-mapped to the default DPR's locations: R240-243 page 21.

Data Page Register relocation is illustrated in Figure 9.

Figure 17. Addressing via CSR, ISR, and DMASR



MMU REGISTERS (Cont'd)

DATA PAGE REGISTER 0 (DPR0)
R240 - Read/Write
Register Page: 21
Reset value: undefined

This register is relocated to R224 if EMR2.5 is set.

7							0
DPR0_7	DPR0_6	DPR0_5	DPR0_4	DPR0_3	DPR0_2	DPR0_1	DPR0_0

Bits 7:0 = **DPR0_[7:0]**: These bits define the 16-Kbyte Data Memory page number. They are used as the most significant address bits (A21-14) to extend the address during a Data Memory access. The DPR0 register is used when addressing the virtual address range 0000h-3FFFh.

DATA PAGE REGISTER 1 (DPR1)
R241 - Read/Write
Register Page: 21
Reset value: undefined

This register is relocated to R225 if EMR2.5 is set.

7							0
DPR1_7	DPR1_6	DPR1_5	DPR1_4	DPR1_3	DPR1_2	DPR1_1	DPR1_0

Bits 7:0 = **DPR1_[7:0]**: These bits define the 16-Kbyte Data Memory page number. They are used as the most significant address bits (A21-14) to extend the address during a Data Memory access. The DPR1 register is used when addressing the virtual address range 4000h-7FFFh.

DATA PAGE REGISTER 2 (DPR2)
R242 - Read/Write
Register Page: 21
Reset value: undefined

This register is relocated to R226 if EMR2.5 is set.

7							0
DPR2_7	DPR2_6	DPR2_5	DPR2_4	DPR2_3	DPR2_2	DPR2_1	DPR2_0

Bits 7:0 = **DPR2_[7:0]**: These bits define the 16-Kbyte Data memory page. They are used as the most significant address bits (A21-14) to extend the address during a Data memory access. The DPR2 register is involved when the virtual address is in the range 8000h-BFFFh.

DATA PAGE REGISTER 3 (DPR3)
R243 - Read/Write
Register Page: 21
Reset value: undefined

This register is relocated to R227 if EMR2.5 is set.

7							0
DPR3_7	DPR3_6	DPR3_5	DPR3_4	DPR3_3	DPR3_2	DPR3_1	DPR3_0

Bits 7:0 = **DPR3_[7:0]**: These bits define the 16-Kbyte Data memory page. They are used as the most significant address bits (A21-14) to extend the address during a Data memory access. The DPR3 register is involved when the virtual address is in the range C000h-FFFFh.

MMU REGISTERS (Cont'd)

2.7.2 CSR: Code Segment Register

This register selects the 64-Kbyte code segment being used at run-time to access instructions. It can also be used to access data if the `spm` instruction has been executed (or `ldpp, ldpd, lddp`). Only the 6 LSBs of the CSR register are implemented, and bits 6 and 7 are reserved. The CSR register allows access to the entire memory space, divided into 64 segments of 64 Kbytes.

To generate the 22-bit Program memory address, the contents of the CSR register is directly used as the 6 MSBs, and the 16-bit virtual address as the 16 LSBs.

Note: The CSR register should only be read and not written for data operations (there are some exceptions which are documented in the following paragraph). It is, however, modified either directly by means of the `jps` and `calls` instructions, or indirectly via the stack, by means of the `rets` instruction.

CODE SEGMENT REGISTER (CSR)
R244 **-** Read/Write
Register Page: 21
Reset value: 0000 0000 (00h)

7							0
0	0	CSR_5	CSR_4	CSR_3	CSR_2	CSR_1	CSR_0

Bits 7:6 = Reserved, keep in reset state.

Bits 5:0 = **CSR_[5:0]**: These bits define the 64-Kbyte memory segment (among 64) which contains the code being executed. These bits are used as the most significant address bits (A21-16).

2.7.3 ISR: Interrupt Segment Register

INTERRUPT SEGMENT REGISTER (ISR)
R248 - Read/Write
Register Page: 21
Reset value: undefined

7							0
0	0	ISR_5	ISR_4	ISR_3	ISR_2	ISR_1	ISR_0

ISR and ENCSR bit (EMR2 register) are also described in the chapter relating to Interrupts, please refer to this description for further details.

Bits 7:6 = Reserved, keep in reset state.

Bits 5:0 = **ISR_[5:0]**: These bits define the 64-Kbyte memory segment (among 64) which contains the interrupt vector table and the code for interrupt service routines and DMA transfers (when the PS bit of the DAPR register is reset). These bits are used as the most significant address bits (A21-16). The ISR is used to extend the address space in two cases:

– Whenever an interrupt occurs: ISR points to the 64-Kbyte memory segment containing the interrupt vector table and the interrupt service routine code. See also the Interrupts chapter.
– During DMA transactions between the peripheral and memory when the PS bit of the DAPR register is reset : ISR points to the 64 K-byte Memory segment that will be involved in the DMA transaction.

2.7.4 DMASR: DMA Segment Register

DMA SEGMENT REGISTER (DMASR)
R249 - Read/Write
Register Page: 21
Reset value: undefined

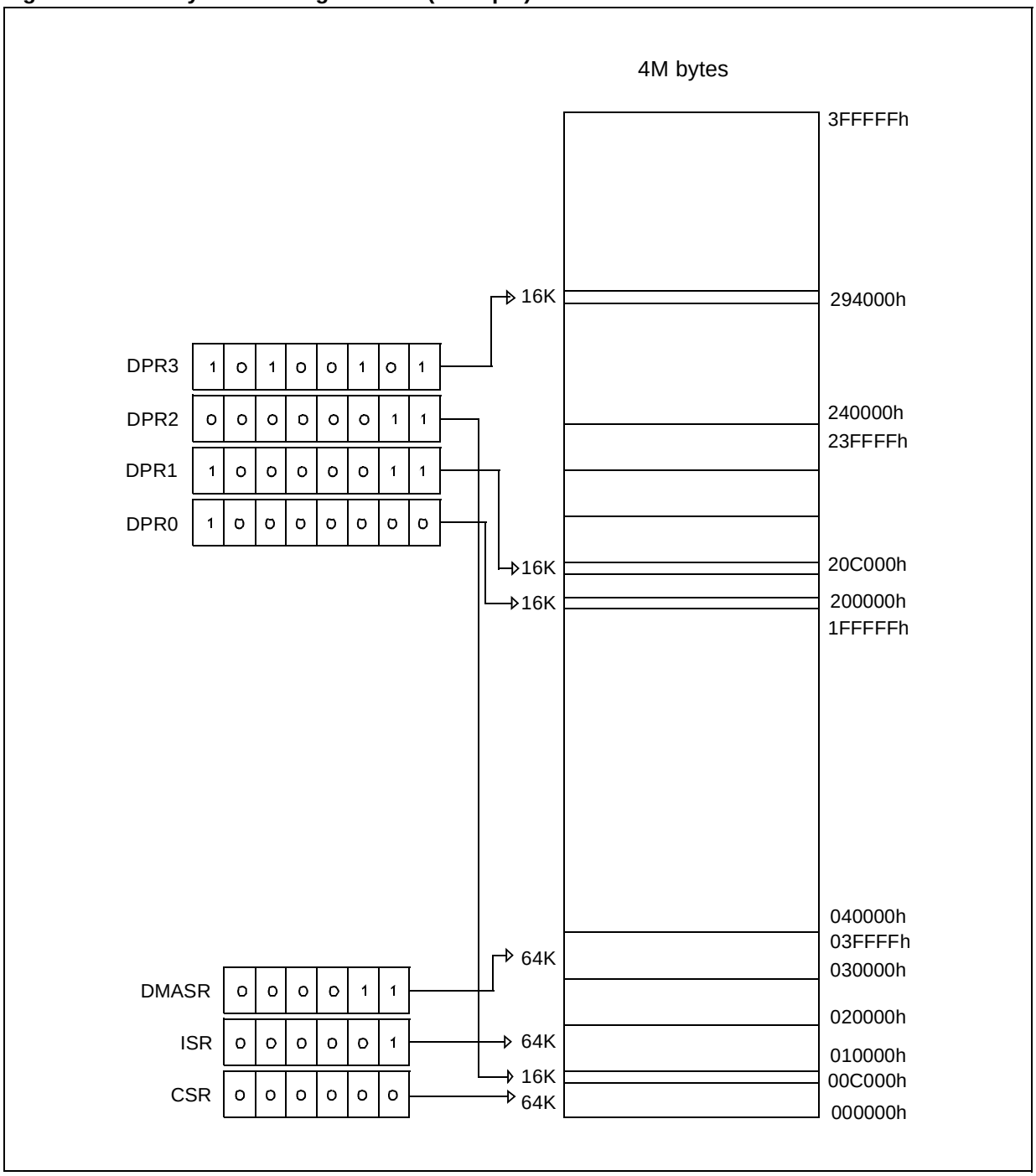
7							0
0	0	DMA SR_5	DMA SR_4	DMA SR_3	DMA SR_2	DMA SR_1	DMA SR_0

Bits 7:6 = Reserved, keep in reset state.

Bits 5:0 = **DMASR_[5:0]**: These bits define the 64-Kbyte Memory segment (among 64) used when a DMA transaction is performed between the peripheral's data register and Memory, with the PS bit of the DAPR register set. These bits are used as the most significant address bits (A21-16). If the PS bit is reset, the ISR register is used to extend the address.

MMU REGISTERS (Cont'd)

Figure 18. Memory Addressing Scheme (example)



2.8 MMU USAGE

2.8.1 Normal Program Execution

Program memory is organized as a set of 64-Kbyte segments. The program can span as many segments as needed, but a procedure cannot stretch across segment boundaries. `jps`, `calls` and `rets` instructions, which automatically modify the CSR, must be used to jump across segment boundaries. Writing to the CSR is forbidden during normal program execution because it is not synchronized with the opcode fetch. This could result in fetching the first byte of an instruction from one memory segment and the second byte from another. Writing to the CSR is allowed when it is not being used, i.e during an interrupt service routine if ENCSR is reset.

Note that a routine must always be called in the same way, i.e. either always with `call` or always with `calls`, depending on whether the routine ends with `ret` or `rets`. This means that if the routine is written without prior knowledge of the location of other routines which call it, and all the program code does not fit into a single 64-Kbyte segment, then `calls`/`rets` should be used.

In typical microcontroller applications, less than 64 Kbytes of RAM are used, so the four Data space pages are normally sufficient, and no change of DPR[3:0] is needed during Program execution. It may be useful however to map part of the ROM into the data space if it contains strings, tables, bit maps, etc.

If there is to be frequent use of paging, the user can set bit 5 (DPRREM) in register R246 (EMR2) of Page 21. This swaps the location of registers DPR[3:0] with that of the data registers of Ports 0-3. In this way, DPR registers can be accessed without the need to save/set/restore the Page Pointer Register. Port registers are therefore moved to page 21. Applications that require a lot of paging typically use more than 64 Kbytes of external memory, and as ports 0, 1 and 2 are required to address it, their data registers are unused.

2.8.2 Interrupts

The ISR register has been created so that the interrupt routines may be found by means of the same vector table even after a segment jump/call.

When an interrupt occurs, the CPU behaves in one of 2 ways, depending on the value of the ENCSR bit in the EMR2 register (R246 on Page 21).

If this bit is reset (default condition), the CPU works in original ST9 compatibility mode. For the duration of the interrupt service routine, the ISR is used instead of the CSR, and the interrupt stack frame is kept exactly as in the original ST9 (only the PC and flags are pushed). This avoids the need to save the CSR on the stack in the case of an interrupt, ensuring a fast interrupt response time. The drawback is that it is not possible for an interrupt service routine to perform segment `calls`/`jps`: these instructions would update the CSR, which, in this case, is not used (ISR is used instead). The code size of all interrupt service routines is thus limited to 64 Kbytes.

If, instead, bit 6 of the EMR2 register is set, the ISR is used only to point to the interrupt vector table and to initialize the CSR at the beginning of the interrupt service routine: the old CSR is pushed onto the stack together with the PC and the flags, and then the CSR is loaded with the ISR. In this case, an `iret` will also restore the CSR from the stack. This approach lets interrupt service routines access the whole 4-Mbyte address space. The drawback is that the interrupt response time is slightly increased, because of the need to also save the CSR on the stack. Compatibility with the original ST9 is also lost in this case, because the interrupt stack frame is different; this difference, however, would not be noticeable for a vast majority of programs.

Data memory mapping is independent of the value of bit 6 of the EMR2 register, and remains the same as for normal code execution: the stack is the same as that used by the main program, as in the ST9. If the interrupt service routine needs to access additional Data memory, it must save one (or more) of the DPRs, load it with the needed memory page and restore it before completion.

2.8.3 DMA

Depending on the PS bit in the DAPR register (see DMA chapter) DMA uses either the ISR or the DMASR for memory accesses: this guarantees that a DMA will always find its memory segment(s), no matter what segment changes the application has performed. Unlike interrupts, DMA transactions cannot save/restore paging registers, so a dedicated segment register (DMASR) has been created. Having only one register of this kind means that all DMA accesses should be programmed in one of the two following segments: the one pointed to by the ISR (when the PS bit of the DAPR register is reset), and the one referenced by the DMASR (when the PS bit is set).

3 INTERRUPTS

3.1 INTRODUCTION

The ST9 responds to peripheral and external events through its interrupt channels. Current program execution can be suspended to allow the ST9 to execute a specific response routine when such an event occurs, providing that interrupts have been enabled, and according to a priority mechanism. If an event generates a valid interrupt request, the current program status is saved and control passes to the appropriate Interrupt Service Routine.

The ST9 CPU can receive requests from the following sources:

– On-chip peripherals

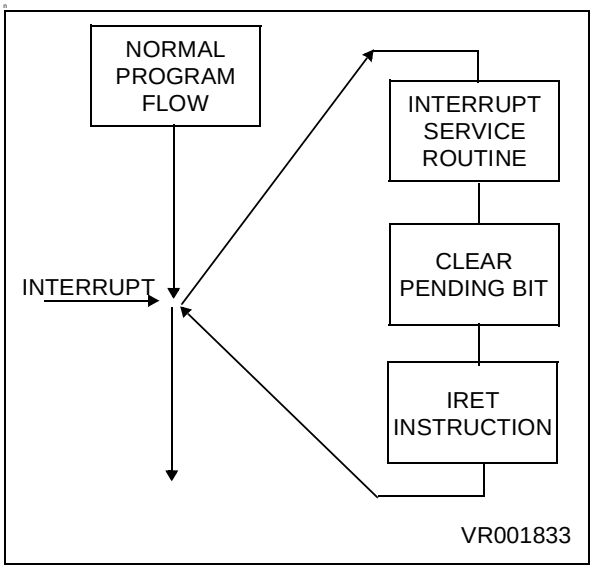
– External pins

– Top-Level Pseudo-non-maskable interrupt

According to the on-chip peripheral features, an event occurrence can generate an Interrupt request which depends on the selected mode.

Up to eight external interrupt channels, with programmable input trigger edge, are available. In addition, a dedicated interrupt channel, set to the Top-level priority, can be devoted either to the external NMI pin (where available) to provide a Non-Maskable Interrupt, or to the Timer/Watchdog. Interrupt service routines are addressed through a vector table mapped in Memory.

Figure 19. Interrupt Response



3.2 INTERRUPT VECTORING

The ST9 implements an interrupt vectoring structure which allows the on-chip peripheral to identify the location of the first instruction of the Interrupt Service Routine automatically.

When an interrupt request is acknowledged, the peripheral interrupt module provides, through its Interrupt Vector Register (IVR), a vector to point into the vector table of locations containing the start addresses of the Interrupt Service Routines (defined by the programmer).

Each peripheral has a specific IVR mapped within its Register File pages.

The Interrupt Vector table, containing the addresses of the Interrupt Service Routines, is located in the first 256 locations of Memory pointed to by the ISR register, thus allowing 8-bit vector addressing. For a description of the ISR register refer to the chapter describing the MMU.

The user Power on Reset vector is stored in the first two physical bytes in memory, 000000h and 000001h.

The Top Level Interrupt vector is located at addresses 0004h and 0005h in the segment pointed to by the Interrupt Segment Register (ISR).

With one Interrupt Vector register, it is possible to address several interrupt service routines; in fact, peripherals can share the same interrupt vector register among several interrupt channels. The most significant bits of the vector are user programmable to define the base vector address within the vector table, the least significant bits are controlled by the interrupt module, in hardware, to select the appropriate vector.

Note: The first 256 locations of the memory segment pointed to by ISR can contain program code.

3.2.1 Divide by Zero trap

The Divide by Zero trap vector is located at addresses 0002h and 0003h of each code segment; it should be noted that for each code segment a Divide by Zero service routine is required.

Warning. *Although the Divide by Zero Trap operates as an interrupt, the FLAG Register is not pushed onto the system Stack automatically. As a result it must be regarded as a subroutine, and the service routine must end with the* `RET` *instruction (not* `IRET` *).*

INTERRUPT VECTORING (Cont'd)

3.2.2 Segment Paging During Interrupt Routines

The ENCSR bit in the EMR2 register can be used to select between original ST9 backward compatibility mode and ST9+ interrupt management mode.

ST9 Backward Compatibility Mode (ENCSR = 0)

If ENCSR is reset, the CPU works in original ST9 compatibility mode. For the duration of the interrupt service routine, ISR is used instead of CSR, and the interrupt stack frame is identical to that of the original ST9: only the PC and Flags are pushed.

This avoids saving the CSR on the stack in the event of an interrupt, thus ensuring a faster interrupt response time.

It is not possible for an interrupt service routine to perform inter-segment calls or jumps: these instructions would update the CSR, which, in this case, is not used (ISR is used instead). The code segment size for all interrupt service routines is thus limited to 64K bytes.

ST9+ Mode (ENCSR = 1)

If ENCSR is set, ISR is only used to point to the interrupt vector table and to initialize the CSR at the beginning of the interrupt service routine: the old CSR is pushed onto the stack together with the PC and flags, and CSR is then loaded with the contents of ISR.

In this case, `iret` will also restore CSR from the stack. This approach allows interrupt service routines to access the entire 4 Mbytes of address space. The drawback is that the interrupt response time is slightly increased, because of the need to also save CSR on the stack.

Full compatibility with the original ST9 is lost in this case, because the interrupt stack frame is different.

ENCSR Bit	0	1
Mode	ST9 Compatible	ST9+
Pushed/Popped Registers	PC, FLAGR	PC, FLAGR, CSR
Max. Code Size for interrupt service routine	64KB Within 1 segment	No limit Across segments

3.3 INTERRUPT PRIORITY LEVELS

The ST9 supports a fully programmable interrupt priority structure. Nine priority levels are available to define the channel priority relationships:

– The on-chip peripheral channels and the eight external interrupt sources can be programmed within eight priority levels. Each channel has a 3-bit field, PRL (Priority Level), that defines its priority level in the range from 0 (highest priority) to 7 (lowest priority).

– The 9th level (Top Level Priority) is reserved for the Timer/Watchdog or the External Pseudo Non-Maskable Interrupt. An Interrupt service routine at this level cannot be interrupted in any arbitration mode. Its mask can be both maskable (TLI) or non-maskable (TLNM).

3.4 PRIORITY LEVEL ARBITRATION

The 3 bits of CPL (Current Priority Level) in the Central Interrupt Control Register contain the priority of the currently running program (CPU priority). CPL is set to 7 (lowest priority) upon reset and can be modified during program execution either by software or automatically by hardware according to the selected Arbitration Mode.

During every instruction, an arbitration phase takes place, during which, for every channel capable of generating an Interrupt, each priority level is compared to all the other requests (interrupts or DMA).

If the highest priority request is an interrupt, its PRL value must be strictly lower (that is, higher priority) than the CPL value stored in the CICR register (R230) in order to be acknowledged. The Top Level Interrupt overrides every other priority.

3.4.1 Priority Level 7 (Lowest)

Interrupt requests at PRL level 7 cannot be acknowledged, as this PRL value (the lowest possible priority) cannot be strictly lower than the CPL value. This can be of use in a fully polled interrupt environment.

3.4.2 Maximum Depth of Nesting

No more than 8 routines can be nested. If an interrupt routine at level N is being serviced, no other Interrupts located at level N can interrupt it. This guarantees a maximum number of 8 nested levels including the Top Level Interrupt request.

PRIORITY LEVEL ARBITRATION (Cont'd)

3.4.3 Simultaneous Interrupts

If two or more requests occur at the same time and at the same priority level, an on-chip daisy chain, specific to every ST9 version, selects the channel with the highest position in the chain, as shown in Table 7. on page 46

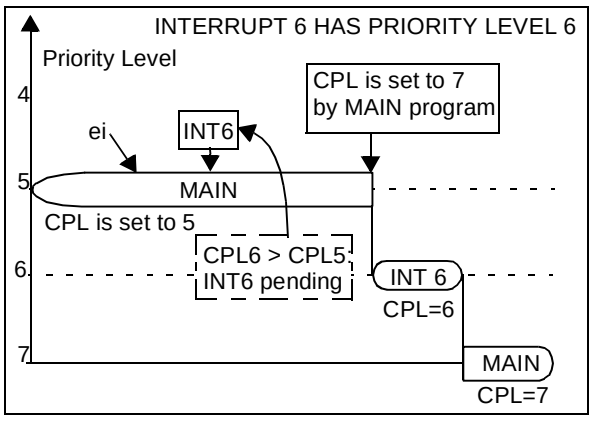
Table 7. Daisy Chain Priority for the ST92195C/D

Highest Position	INTA0	INT0/WDT
	INTA1	Standard Timer 0
	INTB0	INT2/SPI
	INTB1	AD Converter/I²C
	INTC0	INT4/SYNC (EOFVBI)
	INTC1	INT5/SYNC (FLDST)
	INTD0	INT6/Standard Timer 1
Lowest Position	INTD1	INT7

3.4.4 Dynamic Priority Level Modification

The main program and routines can be specifically prioritized. Since the CPL is represented by 3 bits in a read/write register, it is possible to modify dynamically the current priority value during program execution. This means that a critical section can have a higher priority with respect to other interrupt requests. Furthermore it is possible to prioritize even the Main Program execution by modifying the CPL during its execution. See Figure 20 on page 46

Figure 20. Example of Dynamic priority level modification in Nested Mode



3.5 ARBITRATION MODES

The ST9 provides two interrupt arbitration modes: Concurrent mode and Nested mode. Concurrent mode is the standard interrupt arbitration mode. Nested mode improves the effective interrupt response time when service routine nesting is required, depending on the request priority levels.

The IAM control bit in the CICR Register selects Concurrent Arbitration mode or Nested Arbitration Mode.

3.5.1 Concurrent Mode

This mode is selected when the IAM bit is cleared (reset condition). The arbitration phase, performed during every instruction, selects the request with the highest priority level. The CPL value is not modified in this mode.

Start of Interrupt Routine

The interrupt cycle performs the following steps:

– All maskable interrupt requests are disabled by clearing CICR.IEN.

– The PC low byte is pushed onto system stack.

– The PC high byte is pushed onto system stack.

– If ENCSR is set, CSR is pushed onto system stack.

– The Flag register is pushed onto system stack.

– The PC is loaded with the 16-bit vector stored in the Vector Table, pointed to by the IVR.

– If ENCSR is set, CSR is loaded with ISR contents; otherwise ISR is used in place of CSR until `iret` instruction.

End of Interrupt Routine

The Interrupt Service Routine must be ended with the `iret` instruction. The `iret` instruction executes the following operations:

– The Flag register is popped from system stack.

– If ENCSR is set, CSR is popped from system stack.

– The PC high byte is popped from system stack.

– The PC low byte is popped from system stack.

– All unmasked Interrupts are enabled by setting the CICR.IEN bit.

– If ENCSR is reset, CSR is used instead of ISR.

Normal program execution thus resumes at the interrupted instruction. All pending interrupts remain pending until the next `ei` instruction (even if it is executed during the interrupt service routine).

Note: In Concurrent mode, the source priority level is only useful during the arbitration phase, where it is compared with all other priority levels and with the CPL. No trace is kept of its value during the ISR. If other requests are issued during the interrupt service routine, once the global CICR.IEN is re-enabled, they will be acknowledged regardless of the interrupt service routine's priority. This may cause undesirable interrupt response sequences.

ARBITRATION MODES (Cont'd)

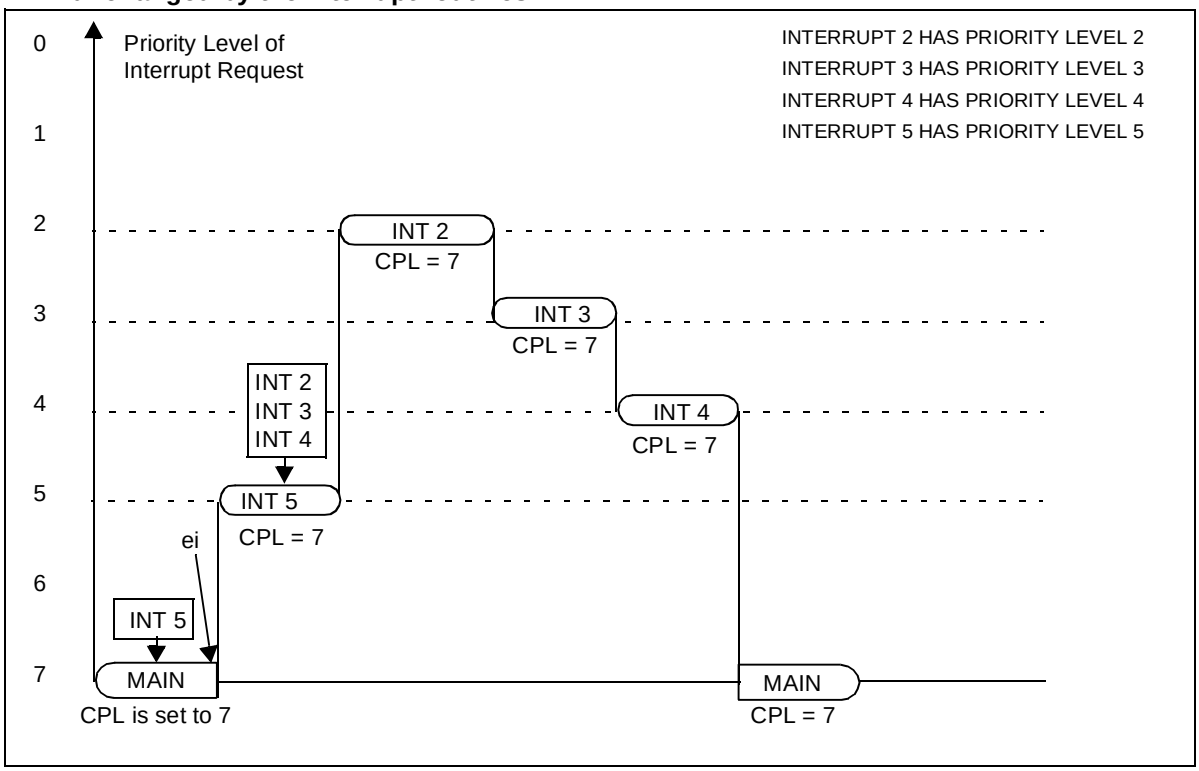
Examples

In the following two examples, three interrupt requests with different priority levels (2, 3 & 4) occur simultaneously during the interrupt 5 service routine.

Example 1

In the first example, (simplest case, Figure 21 on page 47) the `ei` instruction is not used within the interrupt service routines. This means that no new interrupt can be serviced in the middle of the current one. The interrupt routines will thus be serviced one after another, in the order of their priority, until the main program eventually resumes.

Figure 21. Simple Example of a Sequence of Interrupt Requests with:
- Concurrent mode selected and
- IEN unchanged by the interrupt routines



ARBITRATION MODES (Cont'd)

Example 2

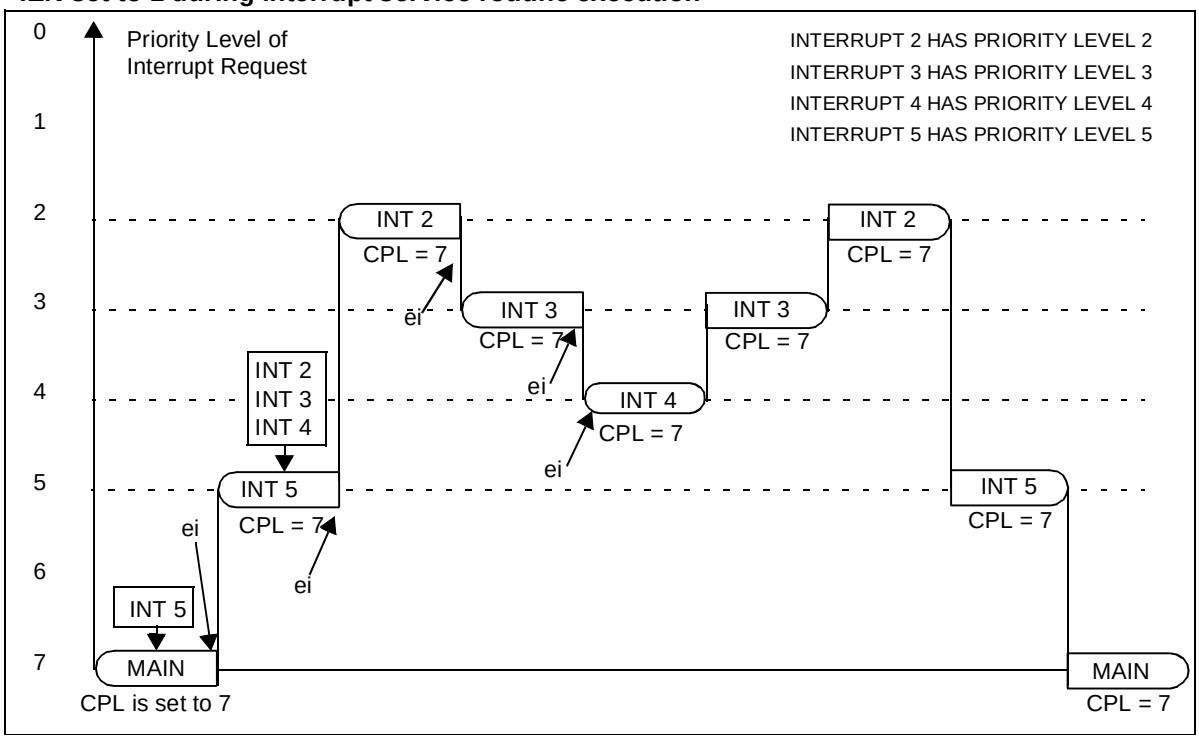
In the second example, (more complex, Figure 22 on page 48), each interrupt service routine sets Interrupt Enable with the `ei` instruction at the beginning of the routine. Placed here, it minimizes response time for requests with a higher priority than the one being serviced.

The level 2 interrupt routine (with the highest priority) will be acknowledged first, then, when the `ei` instruction is executed, it will be interrupted by the level 3 interrupt routine, which itself will be interrupted by the level 4 interrupt routine. When the level 4 interrupt routine is completed, the level 3 interrupt routine resumes and finally the level 2 interrupt routine. This results in the three interrupt service routines being executed in the opposite order of their priority.

It is therefore recommended to avoid inserting the `ei` instruction in the interrupt service routine in Concurrent mode. **Use the `ei` instruction only in nested mode.**

WARNING: If, in Concurrent Mode, interrupts are nested (by executing `ei` in an interrupt service routine), make sure that either ENCSR is set or CSR=ISR, otherwise the `iret` of the innermost interrupt will make the CPU use CSR instead of ISR before the outermost interrupt service routine is terminated, thus making the outermost routine fail.

Figure 22. Complex Example of a Sequence of Interrupt Requests with:
- Concurrent mode selected
- IEN set to 1 during interrupt service routine execution



ARBITRATION MODES (Cont'd)

3.5.2 Nested Mode

The difference between Nested mode and Concurrent mode, lies in the modification of the Current Priority Level (CPL) during interrupt processing.

The arbitration phase is basically identical to Concurrent mode, however, once the request is acknowledged, the CPL is saved in the Nested Interrupt Control Register (NICR) by setting the NICR bit corresponding to the CPL value (i.e. if the CPL is 3, the bit 3 will be set).

The CPL is then loaded with the priority of the request just acknowledged; the next arbitration cycle is thus performed with reference to the priority of the interrupt service routine currently being executed.

Start of Interrupt Routine

The interrupt cycle performs the following steps:

– All maskable interrupt requests are disabled by clearing CICR.IEN.

– CPL is saved in the special NICR stack to hold the priority level of the suspended routine.

– Priority level of the acknowledged routine is stored in CPL, so that the next request priority will be compared with the one of the routine currently being serviced.

– The PC low byte is pushed onto system stack.

– The PC high byte is pushed onto system stack.

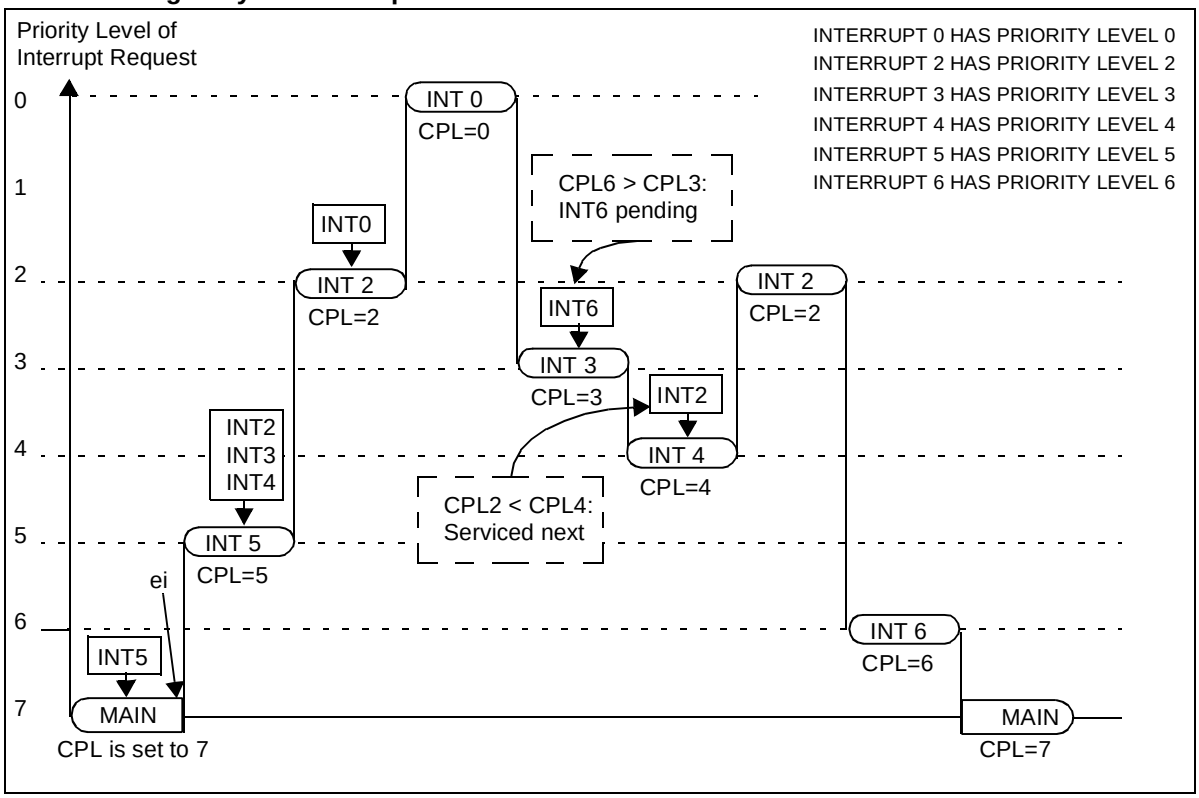
– If ENCSR is set, CSR is pushed onto system stack.

– The Flag register is pushed onto system stack.

– The PC is loaded with the 16-bit vector stored in the Vector Table, pointed to by the IVR.

– If ENCSR is set, CSR is loaded with ISR contents; otherwise ISR is used in place of CSR until `iret` instruction.

Figure 23. Simple Example of a Sequence of Interrupt Requests with:
- Nested mode
- IEN unchanged by the interrupt routines



ARBITRATION MODES (Cont'd)

End of Interrupt Routine

The `iret` Interrupt Return instruction executes the following steps:

– The Flag register is popped from system stack.

– If ENCSR is set, CSR is popped from system stack.

– The PC high byte is popped from system stack.

– The PC low byte is popped from system stack.

– All unmasked Interrupts are enabled by setting the CICR.IEN bit.

– The priority level of the interrupted routine is popped from the special register (NICR) and copied into CPL.

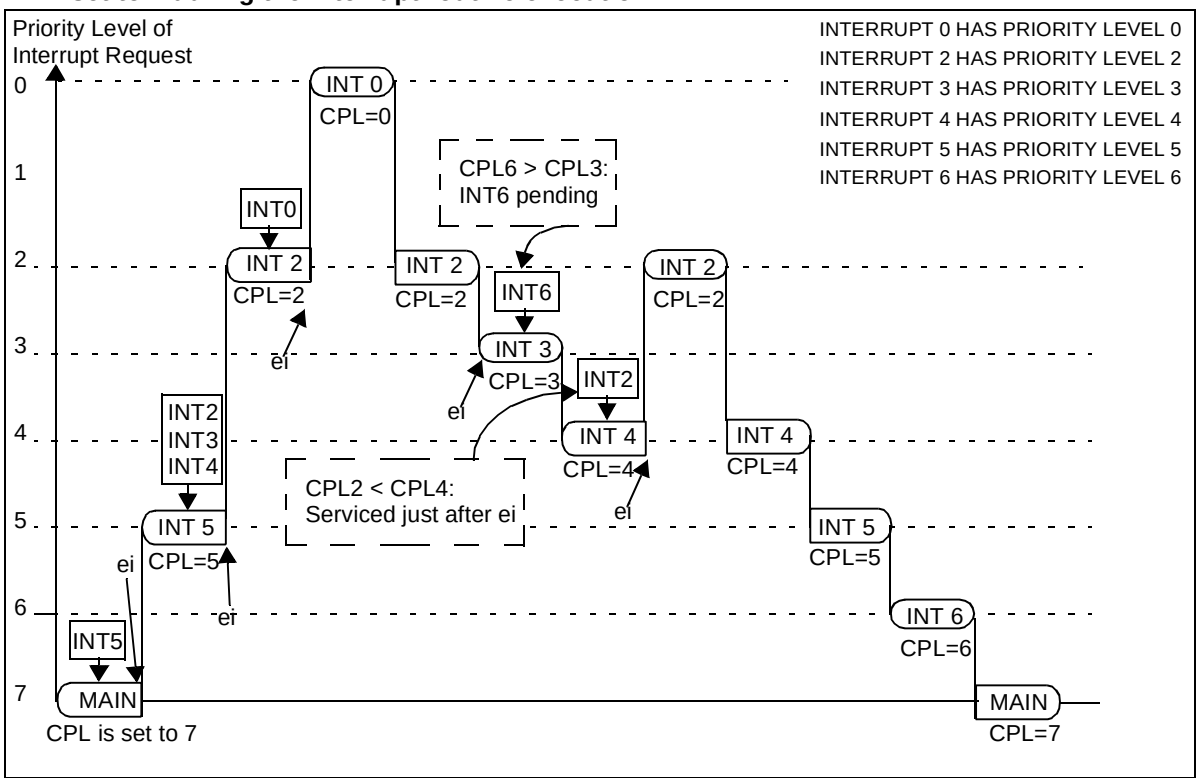
– If ENCSR is reset, CSR is used instead of ISR, unless the program returns to another nested routine.

The suspended routine thus resumes at the interrupted instruction.

Figure 23 on page 49 contains a simple example, showing that if the `ei` instruction is not used in the interrupt service routines, nested and concurrent modes are equivalent.

Figure 24 on page 50 contains a more complex example showing how nested mode allows nested interrupt processing (enabled inside the interrupt service routinesi using the `ei` instruction) according to their priority level.

Figure 24. Complex Example of a Sequence of Interrupt Requests with:
- Nested mode
- IEN set to 1 during the interrupt routine execution



3.6 EXTERNAL INTERRUPTS

The standard ST9 core contains 8 external interrupts sources grouped into four pairs.

Table 8. External Interrupt Channel Grouping

External Interrupt	Channel
INT7 INT6	INTD1 INTD0
INT5 INT4	INTC1 INTC0
INT3 INT2	INTB1 INTB0
INT1 INT0	INTA1 INTA0

Each source has a trigger control bit TEA0,..TED1 (R242,EITR.0,..,7 Page 0) to select triggering on the rising or falling edge of the external pin. If the Trigger control bit is set to “1”, the corresponding pending bit IPA0,..,IPD1 (R243,EIPR.0,..,7 Page 0) is set on the input pin rising edge, if it is cleared, the pending bit is set on the falling edge of the input pin. Each source can be individually masked through the corresponding control bit IMA0,..,IMD1 (EIMR.7,..,0). See Figure 26 on page 53.

The priority level of the external interrupt sources can be programmed among the eight priority levels with the control register EIPLR (R245). The priority level of each pair is software defined using the bits PRL2, PRL1. For each pair, the even channel (A0,B0,C0,D0) of the group has the even priority level and the odd channel (A1,B1,C1,D1) has the odd (lower) priority level.

Figure 25. Priority Level Examples

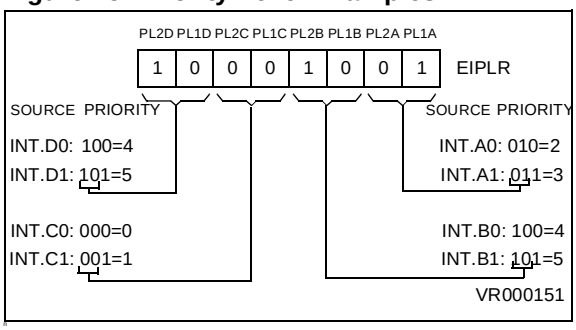


Figure 25 on page 51 shows an example of priority levels.

Figure 26 on page 53 gives an overview of the External interrupt control bits and vectors.

– The source of the interrupt channel INTA0 can be selected between the external pin INT0 (when IA0S = “1”, the reset value) or the On-chip Timer/ Watchdog peripheral (when IA0S = “0”).

– INTA1: by selecting INTS equal to 0, the standard Timer is chosen as the interrupt.

– The source of the interrupt channel INTB0 can be selected between the external pin INT2 (when (SPEN,BMS)=(0,0)) or the SPI peripheral.

– INTB1: setting AD-INT.0 to 1 selects the ADC or I²C as the interrupt source for channel INTB1.

– Setting bit 2 of the CSYCT to 1 selects EOFVBI interrupt as the source for INTC0. Setting this bit to 0 selects external interrupt on INT4.

– Setting FSTEN (bit 3 of the CSYCT register) to 1 selects FLDST interrupt for channel INTC1. Setting this bit to 0 selects external interrupt INT5.

Interrupt channels INTD0 and INTD1 have an input pin as source. However, the input line may be multiplexed with an on-chip peripheral I/O or connected to an input pin that performs also another function.

– Setting the INTS1 bit selects the external interrupt 6 and resetting the INTS1 bit selects the standard timer interrupt.

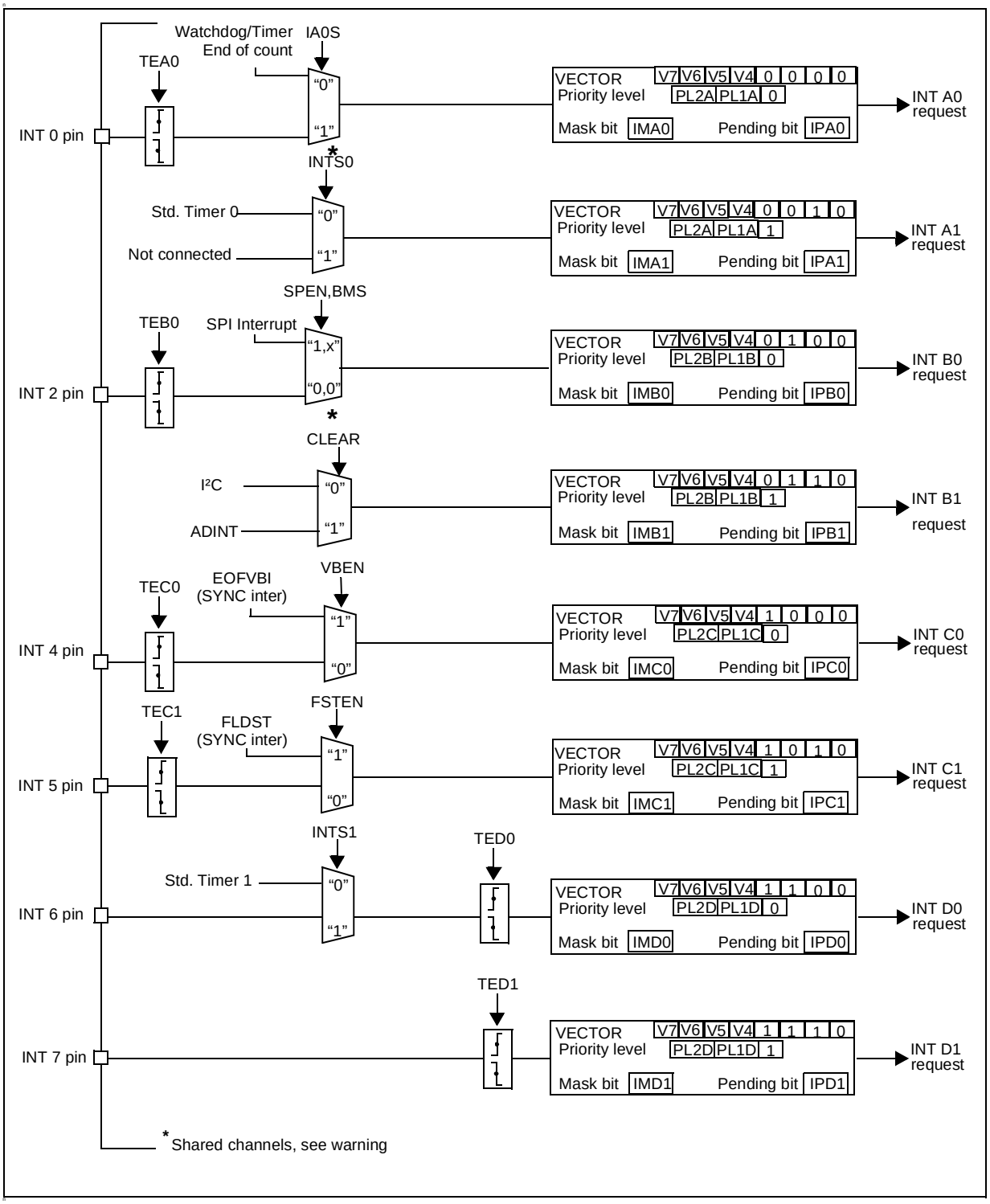
Warning: When using channels shared by both external interrupts and peripherals, special care must be taken to configure their control registers for both peripherals and interrupts.

Table 9. Internal/External Interrupt Source

Channel	Internal Interrupt Source	External Interrupt Source
INTA0	Timer/Watchdog	INT0
INTA1	Standard Timer 0	None
INTB0	SPI Interrupt	INT2
INTB1	A/D Converter / I²C	None
INTC0	EOFVBI (SYNC inter)	INT4
INTC1	FLDST (SYNC inter)	INT5
INTD0	Standard Timer 1	INT6
INTD1	none	INT7

EXTERNAL INTERRUPTS (Cont'd)

Figure 26. External Interrupts Control Bits and Vectors



3.7 TOP LEVEL INTERRUPT

The Top Level Interrupt channel can be assigned either to the external pin NMI or to the Timer/ Watchdog according to the status of the control bit EIVR.TLIS (R246.2, Page 0). If this bit is high (the reset condition) the source is the external pin NMI. If it is low, the source is the Timer/ Watchdog End Of Count. When the source is the NMI external pin, the control bit EIVR.TLTEV (R246.3; Page 0) selects between the rising (if set) or falling (if reset) edge generating the interrupt request. When the selected event occurs, the CICR.TLIP bit (R230.6) is set. Depending on the mask situation, a Top Level Interrupt request may be generated. Two kinds of masks are available, a Maskable mask and a Non-Maskable mask. The first mask is the CICR.TLI bit (R230.5): it can be set or cleared to enable or disable respectively the Top Level Interrupt request. If it is enabled, the global Enable Interrupt bit, CICR.IEN (R230.4) must also be enabled in order to allow a Top Level Request.

The second mask NICR.TLNM (R247.7) is a set-only mask. Once set, it enables the Top Level Interrupt request independently of the value of CICR.IEN and it cannot be cleared by the program. Only the processor RESET cycle can clear this bit. This does not prevent the user from ignoring some sources due to a change in TLIS.

The Top Level Interrupt Service Routine cannot be interrupted by any other interrupt or DMA request, in any arbitration mode, not even by a subsequent Top Level Interrupt request.

Warning. The interrupt machine cycle of the Top Level Interrupt does not clear the CICR.IEN bit, and the corresponding `iret` does not set it. Furthermore the TLI never modifies the CPL bits and the NICR register.

3.8 ON-CHIP PERIPHERAL INTERRUPTS

The general structure of the peripheral interrupt unit is described here, however each on-chip peripheral has its own specific interrupt unit containing one or more interrupt channels, or DMA channels. Please refer to the specific peripheral chapter for the description of its interrupt features and control registers.

The on-chip peripheral interrupt channels provide the following control bits:

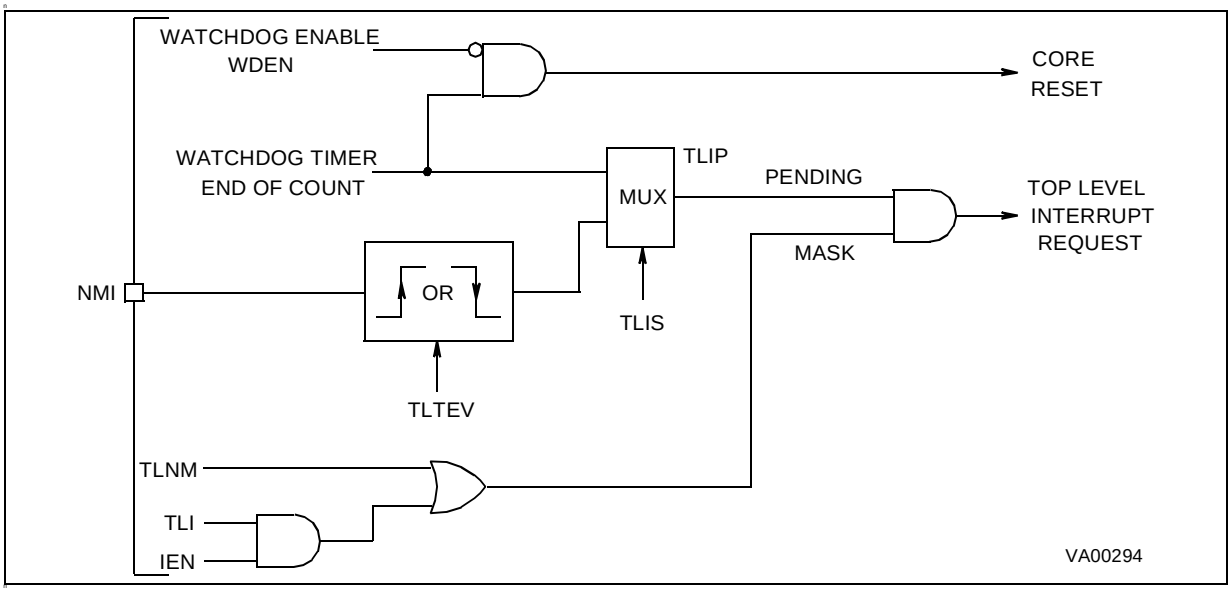
– **Interrupt Pending bit** (IP). Set by hardware when the Trigger Event occurs. Can be set/ cleared by software to generate/cancel pending interrupts and give the status for Interrupt polling.

– **Interrupt Mask bit** (IM). If IM = “0”, no interrupt request is generated. If IM =“1” an interrupt request is generated whenever IP = “1” and CICR.IEN = “1”.

– **Priority Level** (PRL, 3 bits). These bits define the current priority level, PRL=0: the highest priority, PRL=7: the lowest priority (the interrupt cannot be acknowledged)

– **Interrupt Vector Register** (IVR, up to 7 bits). The IVR points to the vector table which itself contains the interrupt routine start address.

Figure 27. Top Level Interrupt Structure



3.9 INTERRUPT RESPONSE TIME

The interrupt arbitration protocol functions completely asynchronously from instruction flow and requires 5 clock cycles. One more CPUCLK cycle is required when an interrupt is acknowledged. Requests are sampled every 5 CPUCLK cycles.

If the interrupt request comes from an external pin, the trigger event must occur a minimum of one INTCLK cycle before the sampling time.

When an arbitration results in an interrupt request being generated, the interrupt logic checks if the current instruction (which could be at any stage of execution) can be safely aborted; if this is the case, instruction execution is terminated immediately and the interrupt request is serviced; if not, the CPU waits until the current instruction is terminated and then services the request. Instruction execution can normally be aborted provided no write operation has been performed.

For an interrupt deriving from an external interrupt channel, the response time between a user event and the start of the interrupt service routine can range from a minimum of 26 clock cycles to a maximum of 55 clock cycles (DIV instruction), 53 clock cycles (DIVWS and MUL instructions) or 49 for other instructions.

For a non-maskable Top Level interrupt, the response time between a user event and the start of the interrupt service routine can range from a minimum of 22 clock cycles to a maximum of 51 clock cycles (DIV instruction), 49 clock cycles (DIVWS and MUL instructions) or 45 for other instructions.

In order to guarantee edge detection, input signals must be kept low/high for a minimum of one INTCLK cycle.

An interrupt machine cycle requires a basic 18 internal clock cycles (CPUCLK), to which must be added a further 2 clock cycles if the stack is in the Register File. 2 more clock cycles must further be added if the CSR is pushed (ENCSR =1).

The interrupt machine cycle duration forms part of the two examples of interrupt response time previously quoted; it includes the time required to push values on the stack, as well as interrupt vector handling.

In Wait for Interrupt mode, a further cycle is required as wake-up delay.

3.10 INTERRUPT REGISTERS

CENTRAL INTERRUPT CONTROL REGISTER (CICR)
R230 - Read/Write
Register Group: System
Reset value: 1000 0111 (87h)

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

Bit 7 = **GCEN**: *Global Counter Enable.*
This bit enables the 16-bit Multifunction Timer peripheral.
0: MFT disabled
1: MFT enabled

Bit 6 = **TLIP**: *Top Level Interrupt Pending.*
This bit is set by hardware when Top Level Interrupt (TLI) trigger event occurs. It is cleared by hardware when a TLI is acknowledged. It can also be set by software to implement a software TLI.
0: No TLI pending
1: TLI pending

Bit 5 = **TLI**: *Top Level Interrupt.*
This bit is set and cleared by software.
0: A Top Level Interrupt is generared when TLIP is set, only if TLNM=1 in the NICR register (independently of the value of the IEN bit).
1: A Top Level Interrupt request is generated when IEN=1 and the TLIP bit are set.

Bit 4 = **IEN**: *Interrupt Enable.*
This bit is cleared by the interrupt machine cycle (except for a TLI).
It is set by the `iret` instruction (except for a return from TLI).
It is set by the `EI` instruction.
It is cleared by the `DI` instruction.
0: Maskable interrupts disabled
1: Maskable Interrupts enabled

Note: The IEN bit can also be changed by software using any instruction that operates on register CICR, however in this case, take care to avoid spurious interrupts, since IEN cannot be cleared in the middle of an interrupt arbitration. Only modify the IEN bit when interrupts are disabled or when no peripheral can generate interrupts. For example, if the state of IEN is not known in advance, and its value must be restored from a previous push of CICR on the stack, use the sequence `DI; POP CICR` to make sure that no interrupts are being arbitrated when CICR is modified.

Bit 3 = **IAM**: *Interrupt Arbitration Mode.*
This bit is set and cleared by software.
0: Concurrent Mode
1: Nested Mode

Bits 2:0 = **CPL[2:0]:** *Current Priority Level.*
These bits define the Current Priority Level. CPL=0 is the highest priority. CPL=7 is the lowest priority. These bits may be modified directly by the interrupt hardware when Nested Interrupt Mode is used.

EXTERNAL INTERRUPT TRIGGER REGISTER (EITR)
R242 - Read/Write
Register Page: 0
Reset value: 0000 0000 (00h)

7							0
TED1	TED0	TEC1	TEC0	TEB1	TEB0	TEA1	TEA0

Bit 7 = **TED1**: *INTD1 Trigger Event*
Bit 6 = **TED0**: *INTD0 Trigger Event*
Bit 5 = **TEC1**: *INTC1 Trigger Event*
Bit 4 = **TEC0**: *INTC0 Trigger Event*
Bit 3 = **TEB1**: *INTB1 Trigger Event*
Bit 2 = **TEB0**: *INTB0 Trigger Event*
Bit 1 = **TEA1**: *INTA1 Trigger Event*
Bit 0 = **TEA0**: *INTA0 Trigger Event*

These bits are set and cleared by software.
0: Select falling edge as interrupt trigger event
1: Select rising edge as interrupt trigger event

INTERRUPT REGISTERS (Cont'd)

EXTERNAL INTERRUPT PENDING REGISTER (EIPR)
R243 - Read/Write
Register Page: 0
Reset value: 0000 0000 (00h)

7							0
IPD1	IPD0	IPC1	IPC0	IPB1	IPB0	IPA1	IPA0

Bit 7 = **IPD1**: *INTD1 Interrupt Pending bit*

Bit 6 = **IPD0**: *INTD0 Interrupt Pending bit*

Bit 5 = **IPC1**: *INTC1 Interrupt Pending bit*

Bit 4 = **IPC0**: *INTC0 Interrupt Pending bit*

Bit 3 = **IPB1**: *INTB1 Interrupt Pending bit*

Bit 2 = **IPB0**: *INTB0 Interrupt Pending bit*

Bit 1 = **IPA1**: *INTA1 Interrupt Pending bit*

Bit 0 = **IPA0**: *INTA0 Interrupt Pending bit*

These bits are set by hardware on occurrence of a trigger event (as specified in the EITR register) and are cleared by hardware on interrupt acknowledge. They can also be set by software to implement a software interrupt.
0: No interrupt pending
1: Interrupt pending

EXTERNAL INTERRUPT MASK-BIT REGISTER (EIMR)
R244 - Read/Write
Register Page: 0
Reset value: 0000 0000 (00h**)**

7							0
IMD1	IMD0	IMC1	IMC0	IMB1	IMB0	IMA1	IMA0

Bit 7 = **IMD1**: *INTD1 Interrupt Mask*

Bit 6 = **IMD0**: *INTD0 Interrupt Mask*

Bit 5 = **IMC1**: *INTC1 Interrupt Mask*

Bit 4 = **IMC0**: *INTC0 Interrupt Mask*

Bit 3 = **IMB1**: *INTB1 Interrupt Mask*

Bit 2 = **IMB0**: *INTB0 Interrupt Mask*

Bit 1 = **IMA1**: *INTA1 Interrupt Mask*

Bit 0 = **IMA0**: *INTA0 Interrupt Mask*

These bits are set and cleared by software.
0: Interrupt masked
1: Interrupt not masked (an interrupt is generated if the IPxx and IEN bits = 1)

EXTERNAL INTERRUPT PRIORITY LEVEL REGISTER (EIPLR)
R245 - Read/Write
Register Page: 0
Reset value: 1111 1111 (FFh**)**

7							0
PL2D	PL1D	PL2C	PL1C	PL2B	PL1B	PL2A	PL1A

Bits 7:6 = **PL2D, PL1D:** *INTD0, D1 Priority Level.*

Bits 5:4 = **PL2C, PL1C**: *INTC0, C1 Priority Level.*

Bits 3:2 = **PL2B, PL1B**: *INTB0, B1 Priority Level.*

Bits 1:0 = **PL2A, PL1A**: *INTA0, A1 Priority Level.*

These bits are set and cleared by software.

The priority is a three-bit value. The LSB is fixed by hardware at 0 for Channels A0, B0, C0 and D0 and at 1 for Channels A1, B1, C1 and D1.

PL2x	PL1x	Hardware bit	Priority
0	0	0 1	0 (Highest) 1
0	1	0 1	2 3
1	0	0 1	4 5
1	1	0 1	6 7 (Lowest)

INTERRUPT REGISTERS (Cont'd)

EXTERNAL INTERRUPT VECTOR REGISTER (EIVR)

R246 - Read/Write
Register Page: 0
Reset value: xxxx 0110b (x6h)

7							0
V7	V6	V5	V4	TLTEV	TLIS	IAOS	EWEN

Bits 7:4 = **V[7:4]**: *Most significant nibble of External Interrupt Vector*.
These bits are not initialized by reset. For a representation of how the full vector is generated from V[7:4] and the selected external interrupt channel, refer to Figure 26 on page 53.

Bit 3 = **TLTEV**: *Top Level Trigger Event bit.*
This bit is set and cleared by software.
0: Select falling edge as NMI trigger event
1: Select rising edge as NMI trigger event

Bit 2 = **TLIS**: *Top Level Input Selection*.
This bit is set and cleared by software.
0: Watchdog End of Count is TL interrupt source
1: NMI is TL interrupt source

Bit 1 = **IA0S**: *Interrupt Channel A0 Selection.*
This bit is set and cleared by software.
0: Watchdog End of Count is INTA0 source
1: External Interrupt pin is INTA0 source

Bit 0 = **EWEN**: *External Wait Enable.*
This bit is set and cleared by software.
0: WAITN pin disabled
1: WAITN pin enabled (to stretch the external memory access cycle).

Note: For more details on Wait mode refer to the section describing the WAITN pin in the External Memory Chapter.

NESTED INTERRUPT CONTROL (NICR)

R247 - Read/Write
Register Page: 0
Reset value: 0000 0000 (00h)

7							0
TLNM	HL6	HL5	HL4	HL3	HL2	HL1	HL0

Bit 7 = **TLNM**: *Top Level Not Maskable*.
This bit is set by software and cleared only by a hardware reset.
0: Top Level Interrupt Maskable. A top level request is generated if the IEN, TLI and TLIP bits =1
1: Top Level Interrupt Not Maskable. A top level request is generated if the TLIP bit =1

Bits 6:0 = **HL[6:0]**: *Hold Level* x
These bits are set by hardware when, in Nested Mode, an interrupt service routine at level x is interrupted from a request with higher priority (other than the Top Level interrupt request). They are cleared by hardware at the `iret` execution when the routine at level x is recovered.

INTERRUPT REGISTERS (Cont'd)

EXTERNAL MEMORY REGISTER 2 (EMR2)

R246 - Read/Write
Register Page: 21
Reset value: 0000 1111 (0Fh)

7							0
0	ENCSR	0	0	1	1	1	1

Bits 7, 5:0 = Reserved, keep in reset state. Refer to the external Memory Interface Chapter.

Bit 6 = **ENCSR**: *Enable Code Segment Register.* This bit is set and cleared by software. It affects the ST9 CPU behaviour whenever an interrupt request is issued.

0: The CPU works in original ST9 compatibility mode. For the duration of the interrupt service routine, ISR is used instead of CSR, and the interrupt stack frame is identical to that of the original ST9: only the PC and Flags are pushed. This avoids saving the CSR on the stack in the event of an interrupt, thus ensuring a faster interrupt response time. The drawback is that it is not possible for an interrupt service routine to perform inter-segment calls or jumps: these instructions would update the CSR, which, in this case, is not used (ISR is used instead). The code segment size for all interrupt service routines is thus limited to 64K bytes.

1: ISR is only used to point to the interrupt vector table and to initialize the CSR at the beginning of the interrupt service routine: the old CSR is pushed onto the stack together with the PC and flags, and CSR is then loaded with the contents of ISR. In this case, `iret` will also restore CSR from the stack. This approach allows interrupt service routines to access the entire 4 Mbytes of address space; the drawback is that the interrupt response time is slightly increased, because of the need to also save CSR on the stack. Full compatibility with the original ST9 is lost in this case, because the interrupt stack frame is different; this difference, however, should not affect the vast majority of programs.

4 RESET AND CLOCK CONTROL UNIT (RCCU)

4.1 INTRODUCTION

The Reset Control Unit comprises two distinct sections:

– An oscillator that uses an external quartz crystal.

– The Reset/Stop Manager, which detects and flags Hardware, Software and Watchdog generated resets.

4.2 RESET / STOP MANAGER

The RESET/STOP Manager resets the device when one of the three following triggering events occurs:

– A hardware reset, consequence of a low level on the **RESET** pin.

– A software reset, consequence of an HALT instruction when enabled.

– A Watchdog end of count.

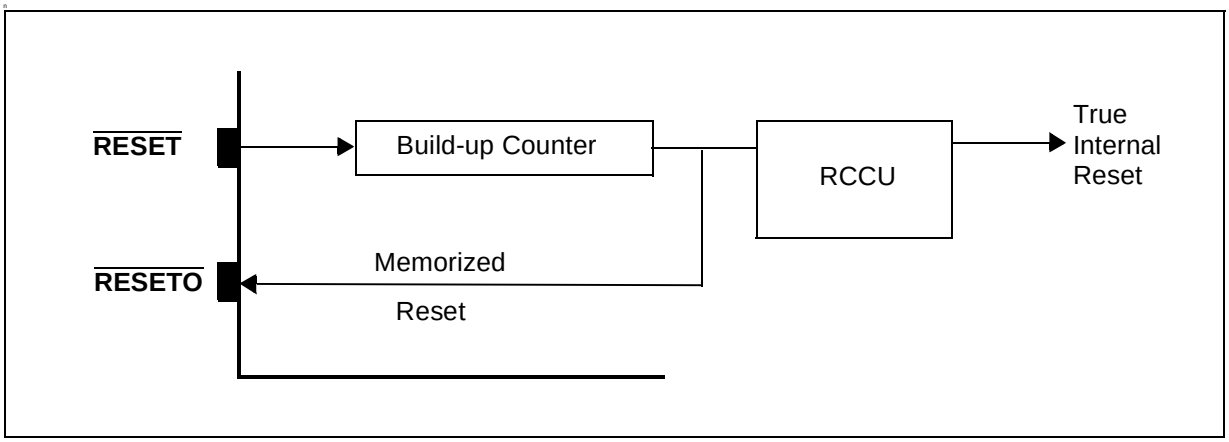
The **RESET** input is schmitt triggered.

Note: The memorized Internal Reset (called RESETO) will be maintained active for a duration of 32768 Oscin periods (about 8 ms for a 4 MHz crystal) after the external input is released (set high).

This RESETO internal Reset signal is output on the I/O port bit P3.7 (active low) during the whole reset phase until the P3.7 configuration is changed by software. The true internal reset (to all macrocells) will only be released 511 Reference clock periods after the Memorized Internal reset is released.

It is possible to know which was the last **RESET** triggering event, by reading bits 5 and 6 of register SDRATH.

Figure 28. Reset Overview



4.3 OSCILLATOR CHARACTERISTICS

The on-chip oscillator circuit uses an inverting gate circuit with tri-state output.

Notes: *Owing to the Q factor required, Ceramic Resonators may not provide a reliable oscillator source*.

The oscillator can not support quartz crystal or ceramic working at the third harmonic without external tank circuits.

OSCOUT must not be used to drive external circuits.

Halt mode is set by means of the HALT instruction. In this mode the parallel resistor, R, is disconnected and the oscillator is disabled. This forces the internal clock to a high level and OSCOUT to a high impedance state.

To exit the HALT condition and restart the oscillator, an external **RESET** pulse is required.

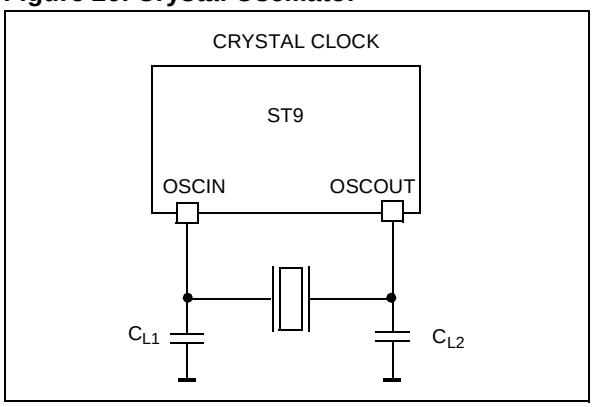
It should be noted that, if the Watchdog function is enabled, a HALT instruction will not disable the oscillator. This to avoid stopping the Watchdog if a HALT code is executed in error. When this occurs, the CPU will be reset when the Watchdog times out or when an external reset is applied.

When a HALT instruction is executed, the main crystal oscillator is stopped and any spurious clocks are ignored. Other analog systems such as the on-chip line PLL (for VPS/WSS) or the whole Video chain (Slicers & Sync Extraction) must be stopped separately by the software as they will induce static consumption.

Table 10. Oscillator Transconductance

gm	Min	Typ	Max
mA/V	0.77	1.5	2.4

Figure 29. Crystal Oscillator



Note: Depending on the application it may be better not to implement CL1.

Figure 30. Internal Oscillator Schematic

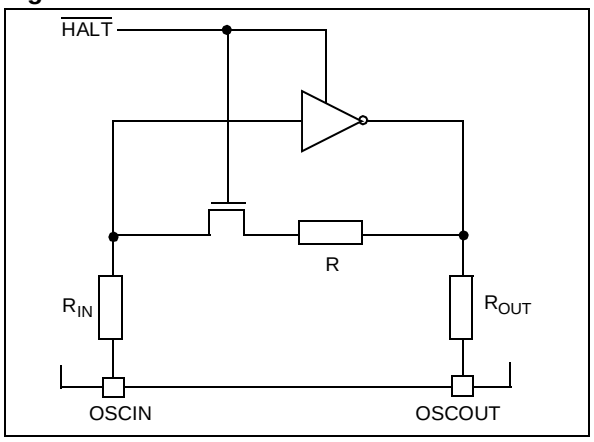
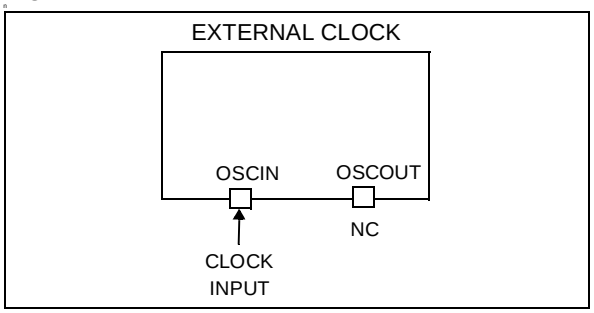


Figure 31. External Clock



The following table is relative to the fundamental quartz crystal only; assuming:

– Rs: parasitic series resistance of the quartz crystal (upper limit)

– C0: parasitic capacitance of the crystal (upper limit, ≤ 7 pF)

– C1,C2: maximum total capacitance on pins OSCIN/OSCOUT (value including external capacitance tied to the pin plus the parasitic capacitance of the board and device).

Table 11. Crystal Specification (C0 ≤ 7 pF)

Freq. MHz.	CL1 = CL2 = 39 pF Rs Max
8	65
4	260

Legend:
Rs: Parasitic Series Resistance of the quartz crystal (upper limit) C0: Parasitic capacitance of the quartz crystal (upper limit, < 7pF)
CL1, CL2: Maximum Total Capacitance on pins OSCIN and OSCOUT (the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device)
gm: Transconductance of the oscillator

Note.The tables are relative to the fundamental quartz crystal only (not ceramic resonator).

4.4 CLOCK CONTROL REGISTERS

MODE REGISTER (MODER)
R235 - Read/Write
Register Group: E (System)
Reset Value: 1110 0000 (E0h)

7							0
1	1	DIV2	PRS2	PRS1	PRS0	0	0

Bits 7:6 = Bits described in Device Architecture chapter.

Bit 5 = **DIV2**: *OSCIN Divided by 2*.
This bit controls the divide by 2 circuit which operates on the OSCIN Clock.
0: No division of OSCIN Clock
1: OSCIN clock is internally divided by 2

Bits 4:2 = **PRS[2:0]**: *Clock Prescaling*.
These bits define the prescaler value used to prescale CPUCLK from INTCLK. When they are reset, the CPUCLK is not prescaled, and is equal to INTCLK; in all other cases, the internal clock is prescaled by the value of these three bits plus one.

Bits 1:0 = Bits described in Device Architecture chapter.

WAIT CONTROL REGISTER (WCR)
R252 - Read/Write
Register Page: 0
Reset Value: 0111 1111 (7Fh)

7							0
0	WDGEN	WDM2	WDM1	WDM0	WPM2	WPM1	WPM0

Bit 7 = Reserved, read as “0”.

Bit 6 = **WDGEN**: refer to Timer/Watchdog chapter.

WARNING. Resetting this bit to zero has the effect of setting the Timer/Watchdog to the Watchdog mode. Unless this is desired, this must be set to “1”.

Bits 5:3 = **WDM[2:0]**: *Data Memory Wait Cycles.*
These bits contain the number of INTCLK cycles to be added automatically to external Data memory accesses. WDM = 0 gives no additional wait cycles. WDM = 7 provides the maximum 7 INTCLK cycles (reset condition).

Bits 2:0 = **WPM[2:0]**: *Program Memory Wait Cycles.*
These bits contain the number of INTCLK cycles to be added automatically to external Program memory accesses. WPM = 0 gives no additional wait cycles, WPM = 7 provides the maximum 7 INTCLK cycles (reset condition).

Note: The number of clock cycles added refers to INTCLK and NOT to CPUCLK.

WARNING. The reset value of the Wait Control Register gives the maximum number of Wait cycles for external memory. To get optimum performance from the ST9 when used in single-chip mode (no external memory) the user should write the WDM2,1,0 and WPM2,1,0 bits to “0”.

4.5 RESET CONTROL UNIT REGISTERS

The RCCU consists of two registers. They are PCONF and SDRATH. Unless otherwise stated, unused register bits must be kept in their reset value in order to avoid problems with the device behaviour.

PLL CONFIGURATION REGISTER (PCONF)
R251 - Read/Write
Register Page: 55
Reset value: 0000 0111 (07h)

7							0
SRESEN	0	0	0	0	1	1	1

Bit 7= **SRESEN**. *Software Reset Enable*.
0: RCCU PLL and CSDU are turned off when a HALT instruction is performed.
1: RCCU will reset the microcontroller when a HALT instruction is performed.

Bits 6:0= Reserved bits. Leave in their reset state.

CLOCK SLOW DOWN UNIT RATIO REGISTER (SDRATH)
R254 - Read/Write
Register Page: 55
Reset value:
0010 0xxx (2xh) after software reset
0100 0xxx (4xh) after watchdog reset
0000 0000 (00h) after external reset

7							0
0	WDGRES	SFTRES	0	0	x	x	x

Bit 7 = Reserved bit. Leave in its reset state.

Bit 6 = **WDGRES**. *Watchdog Reset*. WDGRES is automatically set if the last reset was a watchdog Reset. This is a read only bit.

Bit 5 = **SFTRES**. *Software Reset*. SFTRES is automatically set if the last reset was a software Reset. This is a read only bit.

Bits 4:0 = Reserved bits. Please leave in their reset state.

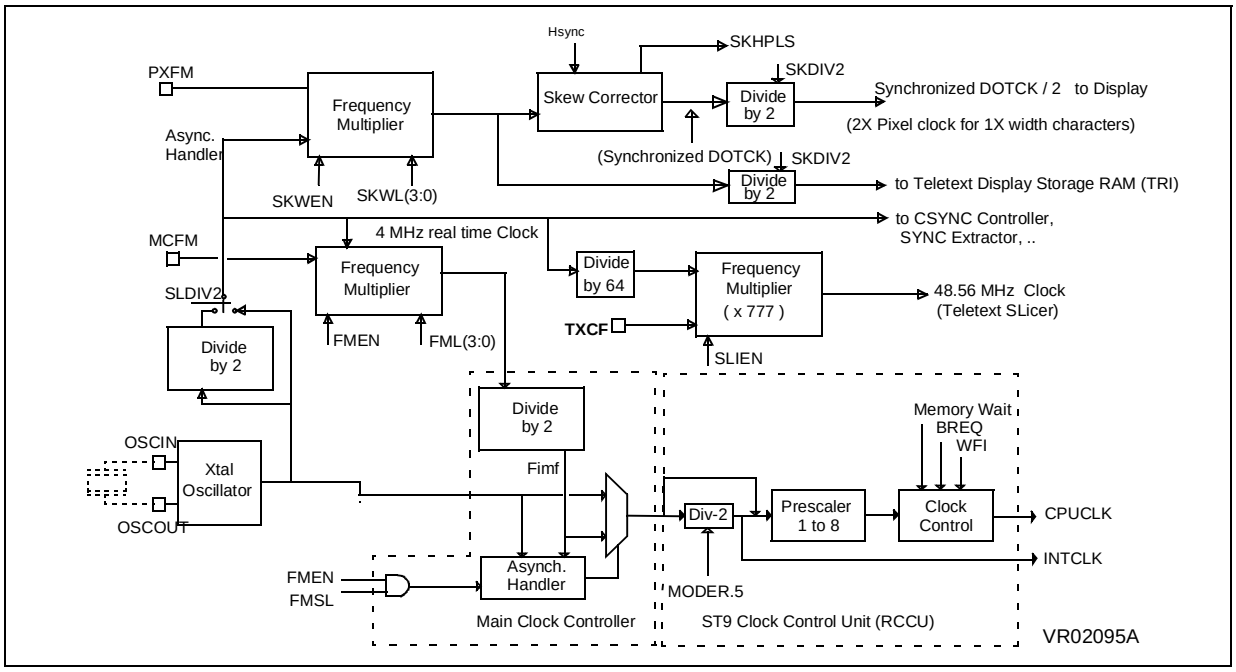
5 TIMING AND CLOCK CONTROLLER

5.1 FREQUENCY MULTIPLIERS

Three on-chip frequency multipliers generate the proper frequencies for: the Core/Real time Peripherals, the Display related time base and the Slicer over-sampling clock for the Teletext Data slicer.

For both the Core and the Display frequency multipliers, a 4 bit programmable feed-back counter allows the adjustment of the multiplying factor to the application need (a 4 MHz or 8 MHz crystal is assumed).

Figure 32. Timing and Clock Controller Block Diagram



FREQUENCY MULTIPLIERS (Cont'd)

For the Off-chip filter components please refer to the Required External Components figure provided in the first section of the data sheet.

The frequency multipliers are off during and upon exiting from the reset phase. The user must program the desired multiplying factor, start the multiplier and then wait for its stability (refer to the Electrical Charateristics chapter for the specified delay).

Once the Core/Peripherals multiplier is stabilized, the Main Clock controller can be re-programmed (through the FMSL bit, MCCR.6) to provide the final frequency (INTCLK) to the CPU.

The frequency multipliers are automatically switched off when the µP enters in HALT mode (the HALT mode forces the control register to its reset status).

Table 12. Examples of CPU speed choice

Crystal Frequency	FML (3:0)	Internal Frequency (Fimf)
4 MHz	4	10 MHz
4 MHz	5	12 MHz
4 MHz	6	14 MHz
4 MHz	7	16 MHz
4 MHz	8	18 MHz
4 MHz	11	24 MHz

Note: 24 MHz is the max. CPU authorized frequency.

Table 13. DOTCK/2 frequency choices

SKW (3:0)	DOTCK/2
8	18 MHz
9	20 MHz(*)
10	22 MHz
11	24 MHz (**)

(*) Preferred values for 4/3.

(**) 16/9 screen formats.

Note: 18 MHz is the min. DOTCK/2 authorized frequency.

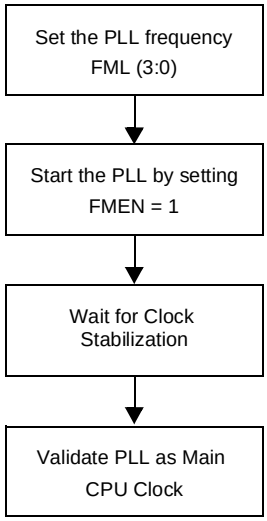
Table 14. Data Slicer over-sampling clock
(other values are not allowed)

Crystal Frequ.	Prescale factor	Multiply factor	7x Text Frequency (Fslic)
4 MHz	64	777	48.5625 MHz
8 MHz	128	777	48.5625 MHz

Table 15. External PLL Filter Stabilisation time

Clock Pin Name	Clock Name	Control Register	Stabilization Period
MCFM	Main Clock PLL Filter Input Pin	MCCR	35 ms.
PXFM	Pixel Clock PLL Filter Input Pin	PXCCR	35 ms
TXCF	Teletext PLL Clock Filter input Pin	SLCCR	200 ms

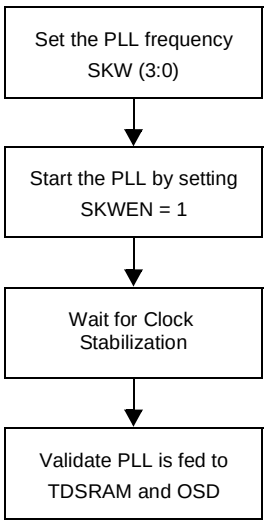
Figure 33. Programming the MCCR



Example:

```
spp        #27h   ;Set the page
ld MCCR,  #04h   ;Set FML bits to the value needed e.g. 10 MHz
or MCCR,  #80h   ;Starts the PLL
 Wait for stabilization time
or MCCR,  #40h   ;Validate the PLL as the main CPU Clock
```

Figure 34. Programming the SKCCR, PXCCR



Example:

```
spp        #27h   ;Set the page
ld SKCCR,  #04h   ;Set SKW bits to the value needed
or SKCCR,  #80h   ;Starts the PLL
 Wait for stabilization time
or PXCCR,  #80h   ;PLL is fed as DOTCK to the TDSRAM & OSDPLL
```

5.2 REGISTER DESCRIPTION

MAIN CLOCK CONTROL REGISTER (MCCR)
R253 - Read/ Write
Register Page: 39
Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
FMEN	FMSL	0	0	FML3	FML2	FML1	FML0

The HALT mode forces the register to its initialization state.

Bit 7 = **FMEN**. *Frequency Multiplier Enable bit.*
0: FM disabled (reset state), low-power consumption mode.
1: FM is enabled, providing clock to the CPU. The FMEN bit must be set only after programming the FML(3:0) bits.

Bit 6= **FMSL**. *Frequency Multiplier Select bit.*
This bit controls the choice of the ST9+ core internal frequency between the external crystal frequency and the Main Clock issued by the frequency multiplier.

In order to secure the application, the ST9+ core internal frequency is automatically switched back to the external crystal frequency if the frequency multiplier is switched off (FMEN =0) regardless of the value of the FMSL bit. Care must be taken to reset the FMSL bit before any frequency multiplier can restart (FMEN set back to 1).

After reset, the external crystal frequency is always sent to the ST9+ Core.

Bits 5:4 = These bits are reserved.

Bits 3:0 = **FML[3:0]** *Frequency bits.*
These 4 bits program the down-counter inserted in the feed-back loop of the Frequency Multiplier which generates the internal multiplied frequency Fimf. The Fimf value is calculated as follows :

Fimf = Crystal frequency * [(FML(3:0) + 1)] /2

SKEW CLOCK CONTROL REGISTER (SKCCR)
R254 - Read/ Write
Register Page: 39
Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SKW EN	SKDIV2	0	0	SKW3	SKW2	SKW1	SKW0

The HALT mode forces the register to its initialization state.

Bit 7= **SKWEN**. *Frequency Multiplier Enable bit.*
0: FM disabled (reset state), low-power consumption mode.
1: FM is enabled, supplying the clock to the Skew corrector. The SKWEN bit must be set only after programming the SKW(3-0) bits.

Bit 6= **SKDIV2**. *Divide-by-2 enable*
This bit determines whether a divide-by-2 downscaling factor is applied to the output of the Skew Corrector.
0 = Divide-by-2 enabled
1 = Divide-by-2 disabled

Bits 5:4 = These bits are reserved.

Bits 3:0 = **SKW[3:0].** *Frequency bits*
These 4 bits program the down-counter inserted in the feedback loop of the Frequency Multiplier which generates the internal multiplied frequency DOTCK. The DOTCK value is calculated as follows :

F(DOTCK) = Crystal frequency * [(SKW(3:0) + 1)]

REGISTER DESCRIPTION (Cont'd)

PLL CLOCK CONTROL REGISTER (**PXCCR)**
R251 - Read/Write
Register Page: 39
Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
PXCE	0	0	0	0	0	0	0

Bit 7= **PXCE**. *Pixel Clock Enable bit.*
0: Pixel and TDSRAM interface clocks are blocked
1: Pixel clock is sent to the display controller and TDSRAM interface.

Bits 6:0= These bits are reserved.

SLICER CLOCK CONTROL REGISTER (**SLCCR)**
R252 - Read/ Write
Register Page: 39
Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SLIEN	SLDIV2	0	VMOD	0	0	0	0

The HALT mode forces the register to its initialization state.

Bit 7= **SLIEN**. *Frequency Multiplier Enable bit.*
0: FM disabled (reset state), low-power consumption mode.
1: FM is enabled, providing clock to the Teletext Data Slicer.

Note: This PLL clock is fed to the VPS/WSS data slicer if the VW_EN bit of the VPSWSSCR register is set.

Bit 6= **SLDIV2**. Divide-by-2 Prescaler.
0: No division, 4 MHz crystal is used
1: Divide-by-2 prescaler enabled. To be used with an 8 MHz crystal.

Bit 5= This bit is reserved.

Bit 4= **VMOD:** *Video mode selection.*
This bit is used to allow a correct teletext slicing depending on the selected video mode (50Hz or 60Hz). It is set and cleared by software.
0: 50 Hz. (41 bytes stored per TV line)
1: 60 Hz (bit rate at 5.733 MHz. and 33 bytes stored per TV line).

Bits 3:0= These bits are reserved.

5.2.1 Register Mapping

The Timing Controller has 4 dedicated registers, mapped in a ST9+ register file page (the page address is 39 (27h)), as follows :

Page 39 (27h)		
FEh	Skew Corrector Control Register	SKCCR
FDh	Main Clock Control Register	MCCR
FCh	SLicer Clock Control Register	SLCCR
FBh	Pixel Clock Control Register	PXCCR

6 I/O PORTS

6.1 INTRODUCTION

ST9 devices feature flexible individually programmable multifunctional input/output lines. Refer to the Pin Description Chapter for specific pin allocations. These lines, which are logically grouped as 8-bit ports, can be individually programmed to provide digital input/output and analog input, or to connect input/output signals to the on-chip peripherals as alternate pin functions. All ports can be individually configured as an input, bi-directional, output or alternate function. In addition, pull-ups can be turned off for open-drain operation, and weak pull-ups can be turned on in their place, to avoid the need for off-chip resistive pull-ups. Ports configured as open drain must never have voltage on the port pin exceeding V_{DD} (refer to the Electrical Characteristics section). Depending on the specific port, input buffers are software selectable to be TTL or CMOS compatible, however on Schmitt trigger ports, no selection is possible.

6.2 SPECIFIC PORT CONFIGURATIONS

Refer to the Pin Description chapter for a list of the specific port styles and reset values.

6.3 PORT CONTROL REGISTERS

Each port is associated with a Data register (PxDR) and three Control registers (PxC0, PxC1, PxC2). These define the port configuration and allow dynamic configuration changes during program execution. Port Data and Control registers are mapped into the Register File as shown in Figure 1. Port Data and Control registers are treated just like any other general purpose register. There are no special instructions for port manipulation: any instruction that can address a register, can address the ports. Data can be directly accessed in the port register, without passing through other memory or “accumulator” locations.

Figure 35. I/O Register Map

	GROUP E			GROUP F PAGE 2	GROUP F PAGE 3	GROUP F PAGE 43	
	System Registers		FFh	Reserved	P7DR	P9DR	R255
			FEh	P3C2	P7C2	P9C2	R254
			FDh	P3C1	P7C1	P9C1	R253
			FCh	P3C0	P7C0	P9C0	R252
			FBh	Reserved	P6DR	P8DR	R251
			FAh	P2C2	P6C2	P8C2	R250
			F9h	P2C1	P6C1	P8C1	R249
			F8h	P2C0	P6C0	P8C0	R248
			F7h	Reserved	Reserved	Reserved	R247
			F6h	P1C2	P5C2		R246
E5h	P5DR	R229	F5h	P1C1	P5C1		R245
E4h	P4DR	R228	F4h	P1C0	P5C0		R244
E3h	P3DR	R227	F3h	Reserved	Reserved		R243
E2h	P2DR	R226	F2h	P0C2	P4C2		R242
E1h	P1DR	R225	F1h	P0C1	P4C1		R241
E0h	P0DR	R224	F0h	P0C0	P4C0		R240

PORT CONTROL REGISTERS (Cont'd)

During Reset, ports with weak pull-ups are set in bidirectional/weak pull-up mode and the output Data Register is set to FFh. This condition is also held after Reset, except for Ports 0 and 1 in ROM-less devices, and can be redefined under software control.

Bidirectional ports without weak pull-ups are set in high impedance during reset. To ensure proper levels during reset, these ports must be externally connected to either V_{DD} or V_{SS} through external pull-up or pull-down resistors.

Other reset conditions may apply in specific ST9 devices.

6.4 INPUT/OUTPUT BIT CONFIGURATION

By programming the control bits PxC0.n and PxC1.n (see Figure 2) it is possible to configure bit Px.n as Input, Output, Bidirectional or Alternate Function Output, where X is the number of the I/O port, and n the bit within the port (n = 0 to 7).

When programmed as input, it is possible to select the input level as TTL or CMOS compatible by programming the relevant PxC2.n control bit.
This option is not available on Schmitt trigger ports.

The output buffer can be programmed as push-pull or open-drain.

A weak pull-up configuration can be used to avoid external pull-ups when programmed as bidirectional (except where the weak pull-up option has been permanently disabled in the pin hardware assignment).

Each pin of an I/O port may assume software programmable Alternate Functions (refer to the device Pin Description and to Section 1.5). To output signals from the ST9 peripherals, the port must be configured as AF OUT. On ST9 devices with A/D Converter(s), configure the ports used for analog inputs as AF IN.

The basic structure of the bit Px.n of a general purpose port Px is shown in Figure 3.

Independently of the chosen configuration, when the user addresses the port as the destination register of an instruction, the port is written to and the data is transferred from the internal Data Bus to the Output Master Latches. When the port is addressed as the source register of an instruction, the port is read and the data (stored in the Input Latch) is transferred to the internal Data Bus.

When Px.n is programmed as an Input:
(See Figure 4).

– The Output Buffer is forced tristate.

– The data present on the I/O pin is sampled into the Input Latch at the beginning of each instruction execution.

– The data stored in the Output Master Latch is copied into the Output Slave Latch at the end of the execution of each instruction. Thus, if bit Px.n is reconfigured as an Output or Bidirectional, the data stored in the Output Slave Latch will be reflected on the I/O pin.

INPUT/OUTPUT BIT CONFIGURATION (Cont'd)

Figure 36. Control Bits

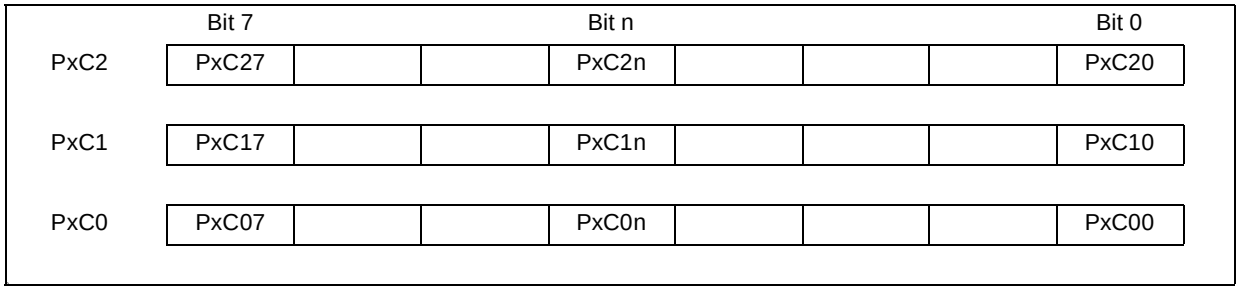


Table 16. Port Bit Configuration Table (n = 0, 1... 7; X = port number)

	General Purpose I/O Pins								A/D Pins
PXC2n PXC1n PXC0n	0 0 0	1 0 0	0 1 0	1 1 0	0 0 1	1 0 1	0 1 1	1 1 1	1 1 1
PXn Configuration	BID	BID	OUT	OUT	IN	IN	AF OUT	AF OUT	AF IN
PXn Output Type	WP OD	OD	PP	OD	HI-Z	HI-Z	PP	OD	HI-Z(1)
PXn Input Type	TTL (or Schmitt Trigger)	TTL (or Schmitt Trigger)	TTL (or Schmitt Trigger)	TTL (or Schmitt Trigger)	CMOS (or Schmitt Trigger)	TTL (or Schmitt Trigger)	TTL (or Schmitt Trigger)	TTL (or Schmitt Trigger)	Analog Input

(1) For A/D Converter inputs.

Legend:

X = Port
n = Bit
AF = Alternate Function
BID = Bidirectional
CMOS= CMOS Standard Input Levels
HI-Z = High Impedance
IN = Input
OD = Open Drain
OUT = Output
PP = Push-Pull
TTL = TTL Standard Input Levels
WP = Weak Pull-up

INPUT/OUTPUT BIT CONFIGURATION (Cont'd)

Figure 37. Basic Structure of an I/O Port Pin

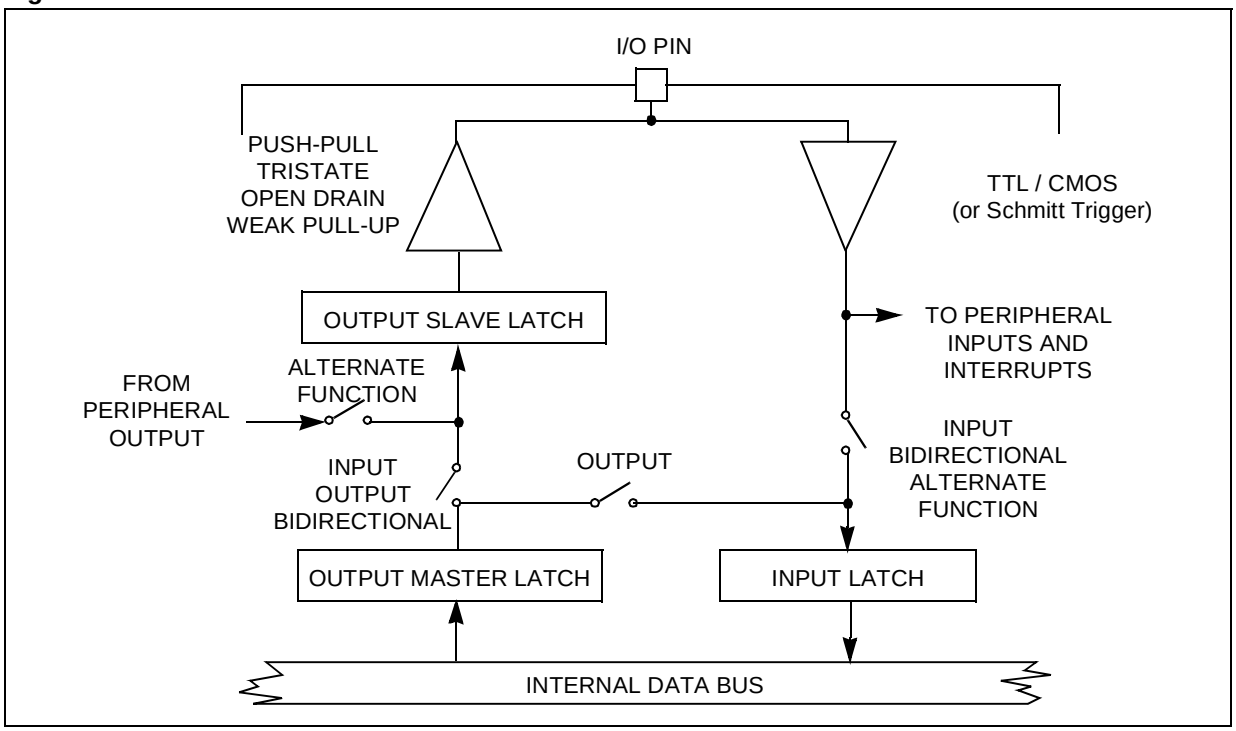


Figure 38. Input Configuration

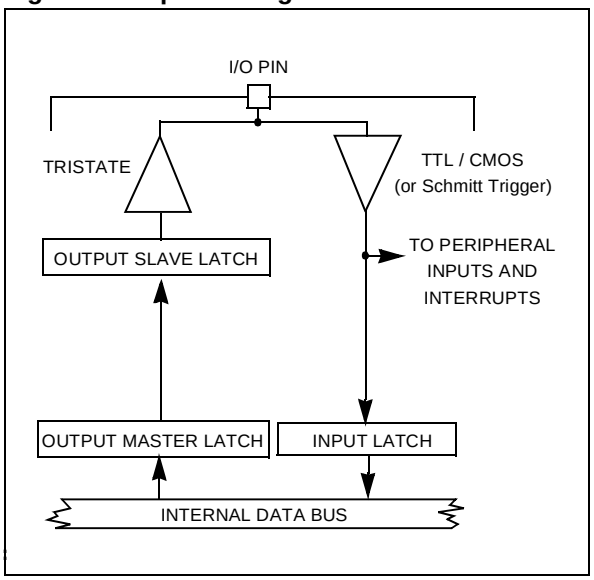
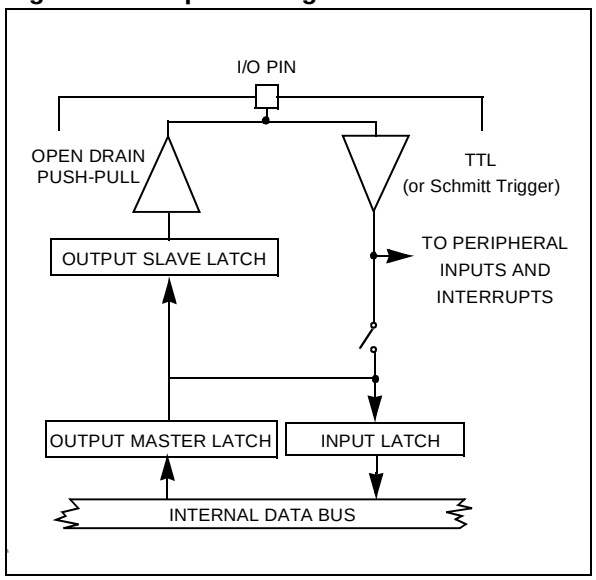


Figure 39. Output Configuration



INPUT/OUTPUT BIT CONFIGURATION (Cont'd)

When Px.n is programmed as an Output:
(Figure 5)

– The Output Buffer is turned on in an Open-drain or Push-pull configuration.

– The data stored in the Output Master Latch is copied both into the Input Latch and into the Output Slave Latch, driving the I/O pin, at the end of the execution of the instruction.

When Px.n is programmed as Bidirectional:
(Figure 6)

– The Output Buffer is turned on in an Open-Drain or Weak Pull-up configuration (except when disabled in hardware).

– The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of the instruction.

– The data stored in the Output Master Latch is copied into the Output Slave Latch, driving the I/O pin, at the end of the execution of the instruction.

WARNING: Due to the fact that in bidirectional mode the external pin is read instead of the output latch, particular care must be taken with arithmetic/logic and Boolean instructions performed on a bidirectional port pin.

These instructions use a read-modify-write sequence, and the result written in the port register depends on the logical level present on the external pin.

This may bring unwanted modifications to the port output register content.

For example:

Port register content, 0Fh
external port value, 03h
(Bits 3 and 2 are externally forced to 0)

A `bset` instruction on bit 7 will return:

Port register content, 83h
external port value, 83h
(Bits 3 and 2 have been cleared).

To avoid this situation, it is suggested that all operations on a port, using at least one bit in bidirectional mode, are performed on a copy of the port register, then transferring the result with a load instruction to the I/O port.

When Px.n is programmed as a digital Alternate Function Output:
(Figure 7)

– The Output Buffer is turned on in an Open-Drain or Push-Pull configuration.

– The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of the instruction.

– The signal from an on-chip function is allowed to load the Output Slave Latch driving the I/O pin. Signal timing is under control of the alternate function. If no alternate function is connected to Px.n, the I/O pin is driven to a high level when in Push-Pull configuration, and to a high impedance state when in open drain configuration.

Figure 40. Bidirectional Configuration

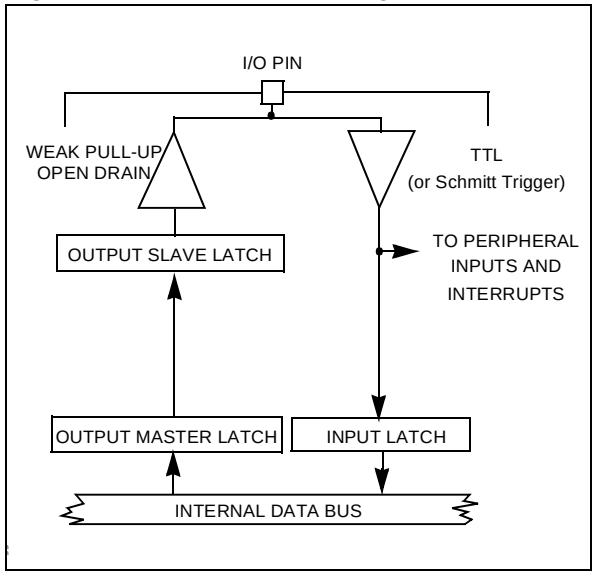
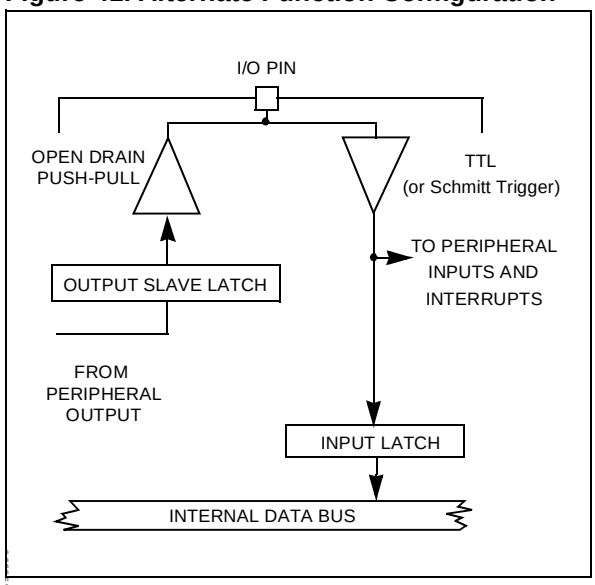


Figure 41. Alternate Function Configuration



6.5 ALTERNATE FUNCTION ARCHITECTURE

Each I/O pin may be connected to three different types of internal signal:

– Data bus Input/Output

– Alternate Function Input

– Alternate Function Output

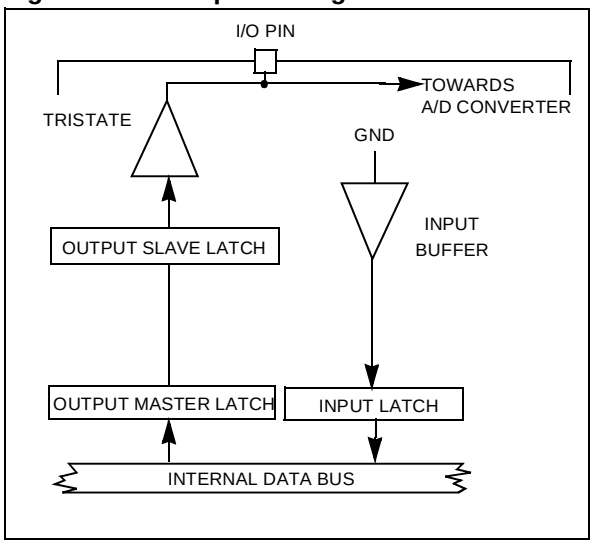
6.5.1 Pin Declared as I/O

A pin declared as I/O, is connected to the I/O buffer. This pin may be an Input, an Output, or a bidirectional I/O, depending on the value stored in (PxC2, PxC1 and PxC0).

6.5.2 Pin Declared as an Alternate Function Input

A single pin may be directly connected to several Alternate Function inputs. In this case, the user must select the required input mode (with the PxC2, PxC1, PxC0 bits) and enable the selected Alternate Function in the Control Register of the peripheral. No specific port configuration is required to enable an Alternate Function input, since the input buffer is directly connected to each alternate function module on the shared pin. As more than one module can use the same input, it is up to the user software to enable the required module as necessary. Parallel I/Os remain operational even when using an Alternate Function input. The exception to this is when an I/O port bit is permanently assigned by hardware as an A/D bit. In this case , after software programming of the bit in AF-OD-TTL, the Alternate function output is forced to logic level 1. The analog voltage level on the corresponding pin is directly input to the A/D (See Figure 8).

Figure 42. A/D Input Configuration



6.5.3 Pin Declared as an Alternate Function Output

The user must select the AF OUT configuration using the PxC2, PxC1, PxC0 bits. Several Alternate Function outputs may drive a common pin. In such case, the Alternate Function output signals are logically ANDed before driving the common pin. The user must therefore enable the required Alternate Function Output by software.

***WARNING**: When a pin is connected both to an alternate function output and to an alternate function input, it should be noted that the output signal will always be present on the alternate function input.*

6.6 I/O STATUS AFTER WFI, HALT AND RESET

The status of the I/O ports during the Wait For Interrupt, Halt and Reset operational modes is shown in the following table. The External Memory Interface ports are shown separately. If only the internal memory is being used and the ports are acting as I/O, the status is the same as shown for the other I/O ports.

Mode	Ext. Mem - I/O Ports		I/O Ports
	P0	P1, P2, P6, P9	
WFI	High Impedance or next address (depending on the last memory operation performed on Port)	Next Address	Not Affected (clock outputs running)
HALT	High Impedance	Next Address	Not Affected (clock outputs stopped)
RESET	Alternate function push-pull (ROMless device)		Bidirectional Weak Pull-up (High impedance when disabled in hardware).

7 ON-CHIP PERIPHERALS

7.1 TIMER/WATCHDOG (WDT)

Important Note: This chapter is a generic description of the WDT peripheral. However depending on the ST9 device, some or all of WDT interface signals described may not be connected to external pins. For the list of WDT pins present on the ST9 device, refer to the device pinout description in the first section of the data sheet.

7.1.1 Introduction

The Timer/Watchdog (WDT) peripheral consists of a programmable 16-bit timer and an 8-bit prescaler. It can be used, for example, to:

– Generate periodic interrupts

– Measure input signal pulse widths

– Request an interrupt after a set number of events

– Generate an output signal waveform

– Act as a Watchdog timer to monitor system integrity

The main WDT registers are:

– Control register for the input, output and interrupt logic blocks (WDTCR)

– 16-bit counter register pair (WDTHR, WDTLR)

– Prescaler register (WDTPR)

The hardware interface consists of up to five signals:

– WDIN External clock input

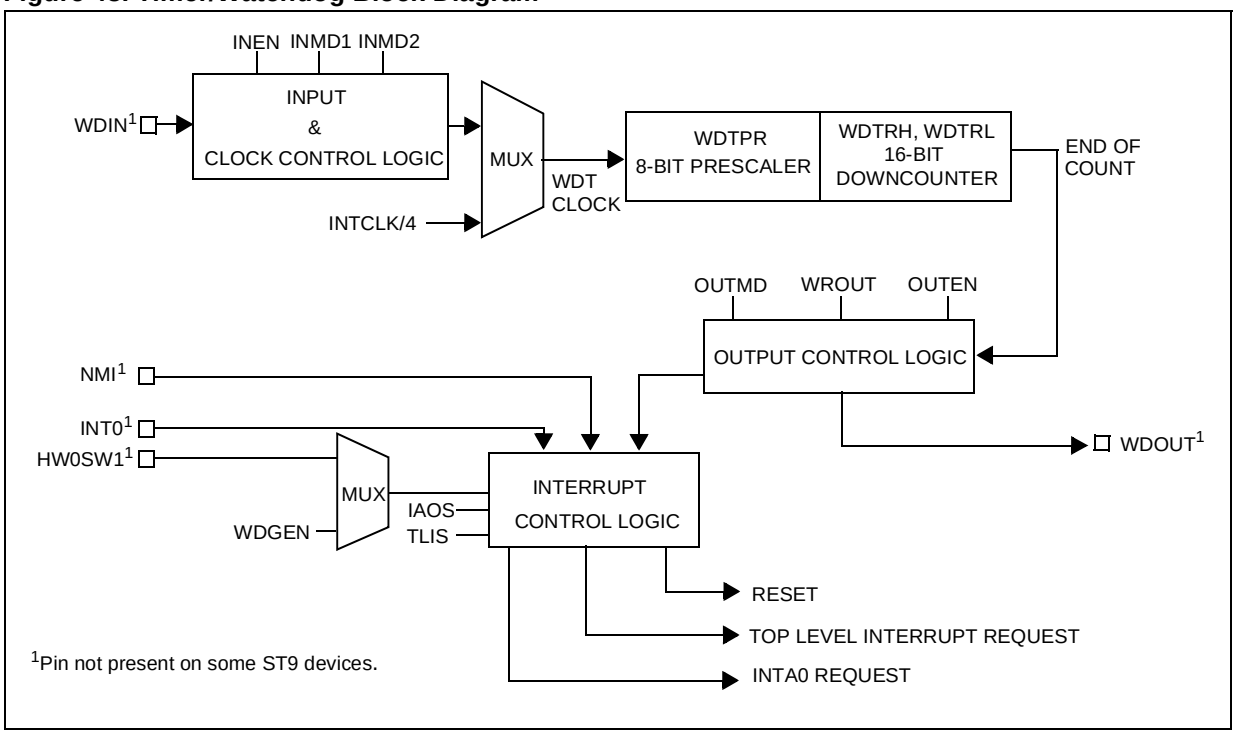
– WDOUT Square wave or PWM signal output

– INT0 External interrupt input

– NMI Non-Maskable Interrupt input

– HW0SW1 Hardware/Software Watchdog enable.

Figure 43. Timer/Watchdog Block Diagram



TIMER/WATCHDOG (Cont'd)

7.1.2 Functional Description

7.1.2.1 External Signals

The HW0SW1 pin can be used to permanently enable Watchdog mode. Refer to section 7.1.3.1 on page 75.

The WDIN Input pin can be used in one of four modes:

– Event Counter Mode

– Gated External Input Mode

– Triggerable Input Mode

– Retriggerable Input Mode

The WDOUT output pin can be used to generate a square wave or a Pulse Width Modulated signal.

An interrupt, generated when the WDT is running as the 16-bit Timer/Counter, can be used as a Top Level Interrupt or as an interrupt source connected to channel A0 of the external interrupt structure (replacing the INT0 interrupt input).

The counter can be driven either by an external clock, or internally by INTCLK divided by 4.

7.1.2.2 Initialisation

The prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be loaded with initial values before starting the Timer/Counter. If this is not done, counting will start with reset values.

7.1.2.3 Start/Stop

The ST_SP bit enables downcounting. When this bit is set, the Timer will start at the beginning of the following instruction. Resetting this bit stops the counter.

If the counter is stopped and restarted, counting will resume from the last value unless a new constant has been entered in the Timer registers (WDTRL, WDTRH).

A new constant can be written in the WDTRH, WDTRL, WDTPR registers while the counter is running. The new value of the WDTRH, WDTRL registers will be loaded at the next End of Count (EOC) condition while the new value of the WDTPR register will be effective immediately.

End of Count is when the counter is 0.

When Watchdog mode is enabled the state of the ST_SP bit is irrelevant.

7.1.2.4 Single/Continuous Mode

The S_C bit allows selection of single or continuous mode.This Mode bit can be written with the Timer stopped or running. It is possible to toggle the S_C bit and start the counter with the same instruction.

Single Mode

On reaching the End Of Count condition, the Timer stops, reloads the constant, and resets the Start/Stop bit. Software can check the current status by reading this bit. To restart the Timer, set the Start/Stop bit.

Note: If the Timer constant has been modified during the stop period, it is reloaded at start time.

Continuous Mode

On reaching the End Of Count condition, the counter automatically reloads the constant and restarts. It is stopped only if the Start/Stop bit is reset.

7.1.2.5 Input Section

If the Timer/Counter input is enabled (INEN bit) it can count pulses input on the WDIN pin. Otherwise it counts the internal clock/4.

For instance, when INTCLK = 24MHz, the End Of Count rate is:

2.79 seconds for Maximum Count
(Timer Const. = FFFFh, Prescaler Const. = FFh)

166 ns for Minimum Count
(Timer Const. = 0000h, Prescaler Const. = 00h)

The Input pin can be used in one of four modes:

– Event Counter Mode

– Gated External Input Mode

– Triggerable Input Mode

– Retriggerable Input Mode

The mode is configurable in the WDTCR.

7.1.2.6 Event Counter Mode

In this mode the Timer is driven by the external clock applied to the input pin, thus operating as an event counter. The event is defined as a high to low transition of the input signal. Spacing between trailing edges should be at least 8 INTCLK periods (or 333ns with INTCLK = 24MHz).

Counting starts at the next input event after the ST_SP bit is set and stops when the ST_SP bit is reset.

TIMER/WATCHDOG (Cont'd)

7.1.2.7 Gated Input Mode

This mode can be used for pulse width measurement. The Timer is clocked by INTCLK/4, and is started and stopped by means of the input pin and the ST_SP bit. When the input pin is high, the Timer counts. When it is low, counting stops. The maximum input pin frequency is equivalent to INTCLK/8.

7.1.2.8 Triggerable Input Mode

The Timer (clocked internally by INTCLK/4) is started by the following sequence:

– setting the Start-Stop bit, followed by

– a High to Low transition on the input pin.

To stop the Timer, reset the ST_SP bit.

7.1.2.9 Retriggerable Input Mode

In this mode, the Timer (clocked internally by INTCLK/4) is started by setting the ST_SP bit. A High to Low transition on the input pin causes counting to restart from the initial value. When the Timer is stopped (ST_SP bit reset), a High to Low transition of the input pin has no effect.

7.1.2.10 Timer/Counter Output Modes

Output modes are selected by means of the OUTEN (Output Enable) and OUTMD (Output Mode) bits of the WDTCR register.

No Output Mode
(OUTEN = "0")

The output is disabled and the corresponding pin is set high, in order to allow other alternate functions to use the I/O pin.

Square Wave Output Mode
(OUTEN = "1", OUTMD = "0")

The Timer outputs a signal with a frequency equal to half the End of Count repetition rate on the WDOUT pin. With an INTCLK frequency of 20MHz, this allows a square wave signal to be generated whose period can range from 400ns to 6.7 seconds.

Pulse Width Modulated Output Mode
(OUTEN = "1", OUTMD = "1")

The state of the WROUT bit is transferred to the output pin (WDOUT) at the End of Count, and is held until the next End of Count condition. The user can thus generate PWM signals by modifying the status of the WROUT pin between End of Count events, based on software counters decremented by the Timer Watchdog interrupt.

7.1.3 Watchdog Timer Operation

This mode is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence of operation. The Watchdog, when enabled, resets the MCU, unless the program executes the correct write sequence before expiry of the programmed time period. The application program must be designed so as to correctly write to the WDTLR Watchdog register at regular intervals during all phases of normal operation.

7.1.3.1 Hardware Watchdog/Software Watchdog

The HW0SW1 pin (when available) selects Hardware Watchdog or Software Watchdog.

If HW0SW1 is held low:

– The Watchdog is enabled by hardware immediately after an external reset. (Note: Software reset or Watchdog reset have no effect on the Watchdog enable status).

– The initial counter value (FFFFh) cannot be modified, however software can change the prescaler value on the fly.

– The WDGEN bit has no effect. (Note: it is not forced low).

If HW0SW1 is held high, or is not present:

– The Watchdog can be enabled by resetting the WDGEN bit.

7.1.3.2 Starting the Watchdog

In Watchdog mode the Timer is clocked by INTCLK/4.

If the Watchdog is software enabled, the time base must be written in the timer registers before entering Watchdog mode by resetting the WDGEN bit. Once reset, this bit cannot be changed by software.

If the Watchdog is hardware enabled, the time base is fixed by the reset value of the registers.

Resetting WDGEN causes the counter to start, regardless of the value of the Start-Stop bit.

In Watchdog mode, only the Prescaler Constant may be modified.

If the End of Count condition is reached a System Reset is generated.

TIMER/WATCHDOG (Cont'd)

7.1.3.3 Preventing Watchdog System Reset

In order to prevent a system reset, the sequence AAh, 55h must be written to WDTLR (Watchdog Timer Low Register). Once 55h has been written, the Timer reloads the constant and counting restarts from the preset value.

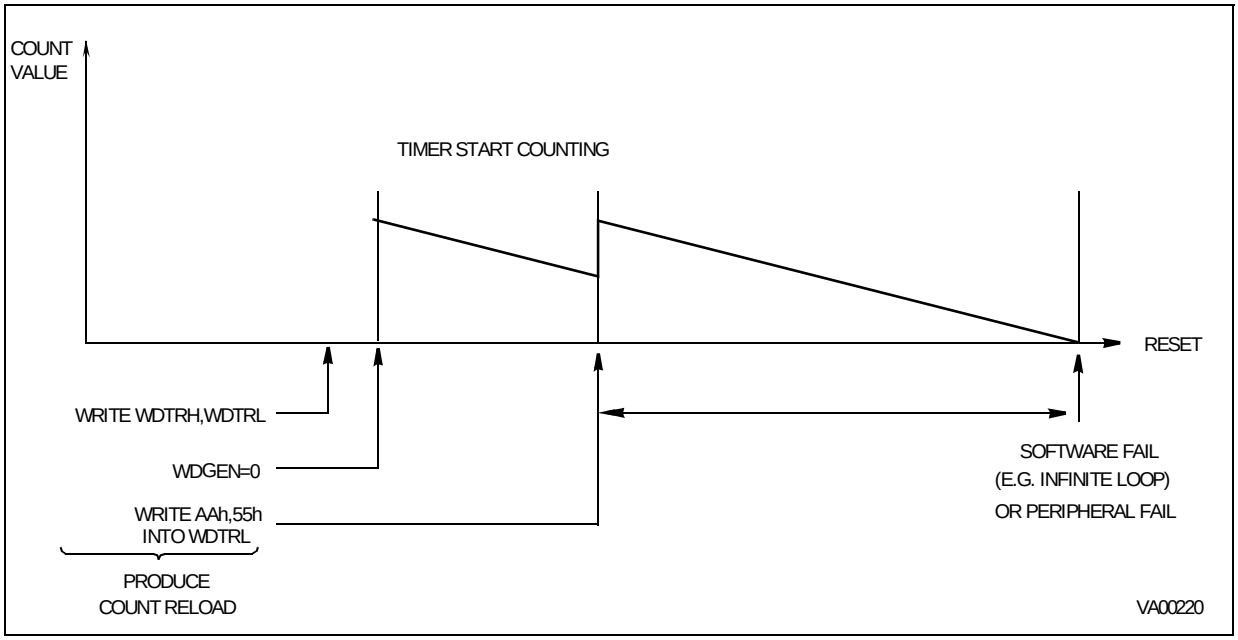
To reload the counter, the two writing operations must be performed sequentially without inserting other instructions that modify the value of the WDTLR register between the writing operations. The maximum allowed time between two reloads of the counter depends on the Watchdog timeout period.

7.1.3.4 Non-Stop Operation

In Watchdog Mode, a `Halt` instruction is regarded as illegal. Execution of the `Halt` instruction stops further execution by the CPU and interrupt acknowledgment, but does not stop INTCLK, CPUCLK or the Watchdog Timer, which will cause a System Reset when the End of Count condition is reached. Furthermore, ST_SP, S_C and the Input Mode selection bits are ignored. Hence, regardless of their status, the counter always runs in Continuous Mode, driven by the internal clock.

The Output mode should not be enabled, since in this context it is meaningless.

Figure 44. Watchdog Timer Mode



TIMER/WATCHDOG (Cont'd)

7.1.4 WDT Interrupts

The Timer/Watchdog issues an interrupt request at every End of Count, when this feature is enabled.

A pair of control bits, IA0S (EIVR.1, Interrupt A0 selection bit) and TLIS (EIVR.2, Top Level Input Selection bit) allow the selection of 2 interrupt sources (Timer/Watchdog End of Count, or External Pin) handled in two different ways, as a Top Level Non Maskable Interrupt (Software Reset), or as a source for channel A0 of the external interrupt logic.

A block diagram of the interrupt logic is given in Figure 44.

Note: Software traps can be generated by setting the appropriate interrupt pending bit.

Table 17 Interrupt Configuration below, shows all the possible configurations of interrupt/reset sources which relate to the Timer/Watchdog.

A reset caused by the watchdog will set bit 6, WDGRES of R242 - Page 55 (Clock Flag Register). See section CLOCK CONTROL REGISTERS.

Figure 45. Interrupt Sources

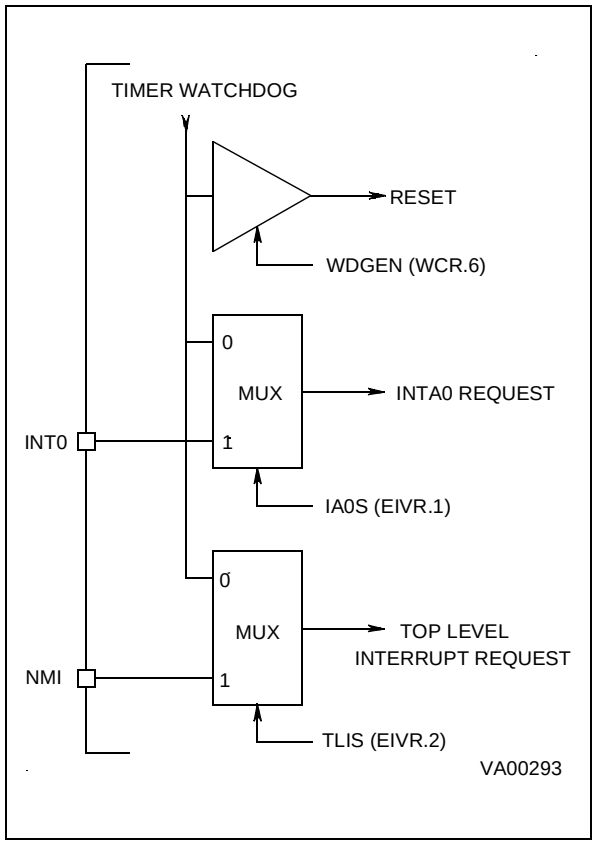


Table 17. Interrupt Configuration

Control Bits			Enabled Sources			Operating Mode
WDGEN	IA0S	TLIS	Reset	INTA0	Top Level	
0	0	0	WDG/Ext Reset	SW TRAP	SW TRAP	Watchdog
0	0	1	WDG/Ext Reset	SW TRAP	Ext Pin	Watchdog
0	1	0	WDG/Ext Reset	Ext Pin	SW TRAP	Watchdog
0	1	1	WDG/Ext Reset	Ext Pin	Ext Pin	Watchdog
1	0	0	Ext Reset	Timer	Timer	Timer
1	0	1	Ext Reset	Timer	Ext Pin	Timer
1	1	0	Ext Reset	Ext Pin	Timer	Timer
1	1	1	Ext Reset	Ext Pin	Ext Pin	Timer

Legend:
WDG = Watchdog function
SW TRAP = Software Trap

Note: If IA0S and TLIS = 0 (enabling the Watchdog EOC as interrupt source for both Top Level and INTA0 interrupts), only the INTA0 interrupt is taken into account.

TIMER/WATCHDOG (Cont'd)

7.1.5 Register Description

The Timer/Watchdog is associated with 4 registers mapped into Group F, Page 0 of the Register File.

WDTHR: Timer/Watchdog High Register

WDTLR: Timer/Watchdog Low Register

WDTPR: Timer/Watchdog Prescaler Register

WDTCR: Timer/Watchdog Control Register

Three additional control bits are mapped in the following registers on Page 0:

Watchdog Mode Enable, (WCR.6)

Top Level Interrupt Selection, (EIVR.2)

Interrupt A0 Channel Selection, (EIVR.1)

Note: The registers containing these bits also contain other functions. Only the bits relevant to the operation of the Timer/Watchdog are shown here.

Counter Register

This 16-bit register (WDTLR, WDTHR) is used to load the 16-bit counter value. The registers can be read or written "on the fly".

TIMER/WATCHDOG HIGH REGISTER (WDTHR)
R248 - Read/Write
Register Page: 0
Reset value: 1111 1111 (FFh)

7							0
R15	R14	R13	R12	R11	R10	R9	R8

Bits 7:0 = **R[15:8]** *Counter Most Significant Bits*.

TIMER/WATCHDOG LOW REGISTER (WDTLR)
R249 - Read/Write
Register Page: 0
Reset value: 1111 1111b (FFh)

7							0
R7	R6	R5	R4	R3	R2	R1	R0

Bits 7:0 = **R[7:0]** *Counter Least Significant Bits.*

TIMER/WATCHDOG PRESCALER REGISTER (WDTPR)
R250 - Read/Write
Register Page: 0
Reset value: 1111 1111 (FFh)

7							0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

Bits 7:0 = **PR[7:0]** *Prescaler value.*
A programmable value from 1 (00h) to 256 (FFh).

Warning: *In order to prevent incorrect operation of the Timer/Watchdog, the prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be initialised before starting the Timer/Watchdog. If this is not done, counting will start with the reset (un-initialised) values.*

WATCHDOG TIMER CONTROL REGISTER (WDTCR)
R251- Read/Write
Register Page: 0
Reset value: 0001 0010 (12h)

7							0
ST_SP	S_C	INMD1	INMD2	INEN	OUTMD	WROUT	OUTEN

Bit 7 = **ST_SP**: *Start/Stop Bit*.
This bit is set and cleared by software.
0: Stop counting
1: Start counting (see Warning above)

Bit 6 = **S_C**: *Single/Continuous*.
This bit is set and cleared by software.
0: Continuous Mode
1: Single Mode

Bits 5:4 = **INMD[1:2]**: *Input mode selection bits*.

These bits select the input mode:

INMD1	INMD2	INPUT MODE
0	0	Event Counter
0	1	Gated Input (Reset value)
1	0	Triggerable Input
1	1	Retriggerable Input

TIMER/WATCHDOG (Cont'd)

Bit 3 = **INEN**: *Input Enable*.
This bit is set and cleared by software.
0: Disable input section
1: Enable input section

Bit 2 = **OUTMD**: *Output Mode.*
This bit is set and cleared by software.
0: The output is toggled at every End of Count
1: The value of the WROUT bit is transferred to the output pin on every End Of Count if OUTEN=1.

Bit 1 = **WROUT**: *Write Out*.
The status of this bit is transferred to the Output pin when OUTMD is set; it is user definable to allow PWM output (on Reset WROUT is set).

Bit 0 = **OUTEN**: *Output Enable bit*.
This bit is set and cleared by software.
0: Disable output
1: Enable output

WAIT CONTROL REGISTER (WCR)
R252 - Read/Write
Register Page: 0
Reset value: 0111 1111 (7Fh)

7							0
x	WDGEN	x	x	x	x	x	x

Bit 6 = **WDGEN**: *Watchdog Enable* (active low).
Resetting this bit via software enters the Watchdog mode. Once reset, it cannot be set anymore by the user program. At System Reset, the Watchdog mode is disabled.

Note: This bit is ignored if the Hardware Watchdog option is enabled by pin HW0SW1 (if available).

EXTERNAL INTERRUPT VECTOR REGISTER (EIVR)
R246 - Read/Write
Register Page: 0
Reset value: xxxx 0110 (x6h)

7							0
x	x	x	x	x	TLIS	IA0S	x

Bit 2 = **TLIS**: *Top Level Input Selection*.
This bit is set and cleared by software.
0: Watchdog End of Count is TL interrupt source
1: NMI is TL interrupt source

Bit 1 = **IA0S**: *Interrupt Channel A0 Selection.*
This bit is set and cleared by software.
0: Watchdog End of Count is INTA0 source
1: External Interrupt pin is INTA0 source

Warning: *To avoid spurious interrupt requests, the IA0S bit should be accessed only when the interrupt logic is disabled (i.e. after the DI instruction). It is also necessary to clear any possible interrupt pending requests on channel A0 before enabling this interrupt channel. A delay instruction (e.g. a NOP instruction) must be inserted between the reset of the interrupt pending bit and the IA0S write instruction.*

Other bits are described in the Interrupt section.

7.2 STANDARD TIMER (STIM)

Important Note: This chapter is a generic description of the STIM peripheral. Depending on the ST9 device, some or all of the interface signals described may not be connected to external pins. For the list of STIM pins present on the particular ST9 device, refer to the pinout description in the first section of the data sheet.

7.2.1 Introduction

The Standard Timer includes a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes capability. The Standard Timer uses an input pin (STIN) and an output (STOUT) pin. These pins, when available, may be independent pins or connected as Alternate Functions of an I/O port bit.

STIN can be used in one of four programmable input modes:

– event counter,

– gated external input mode,

– triggerable input mode,

– retriggerable input mode.

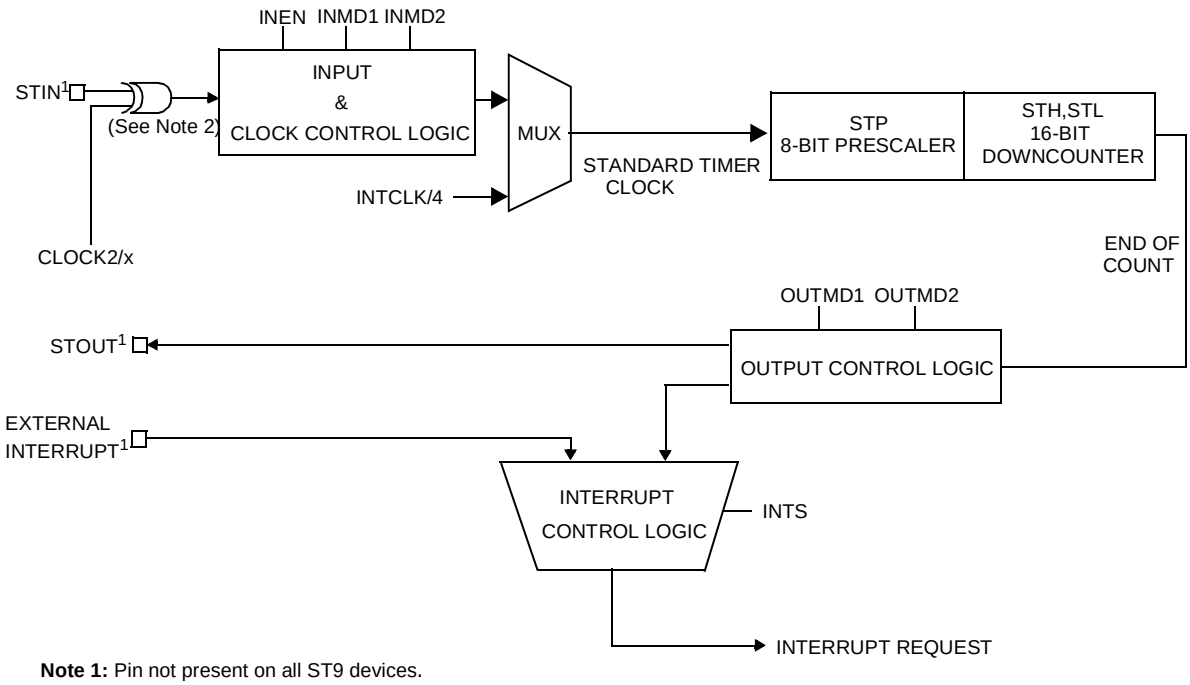
STOUT can be used to generate a Square Wave or Pulse Width Modulated signal.

The Standard Timer is composed of a 16-bit down counter with an 8-bit prescaler. The input clock to the prescaler can be driven either by an internal clock equal to INTCLK divided by 4, or by CLOCK2 derived directly from the external oscillator, divided by device dependent prescaler value, thus providing a stable time reference independent from the PLL programming or by an external clock connected to the STIN pin.

The Standard Timer End Of Count condition is able to generate an interrupt which is connected to one of the external interrupt channels.

The End of Count condition is defined as the Counter Underflow, whenever 00h is reached.

Figure 46. Standard Timer Block Diagram



Note 1: Pin not present on all ST9 devices.

Note 2: Depending on device, the source of the INPUT & CLOCK CONTROL LOGIC block may be permanently connected either to STIN or the RCCU signal CLOCK2/x. In devices without STIN and CLOCK2, the INEN bit must be held at 0.

STANDARD TIMER (Cont'd)

7.2.2 Functional Description

7.2.2.1 Timer/Counter control

Start-stop Count. The ST-SP bit (STC.7) is used in order to start and stop counting. An instruction which sets this bit will cause the Standard Timer to start counting at the beginning of the next instruction. Resetting this bit will stop the counter.

If the counter is stopped and restarted, counting will resume from the value held at the stop condition, unless a new constant has been entered in the Standard Timer registers during the stop period. In this case, the new constant will be loaded as soon as counting is restarted.

A new constant can be written in STH, STL, STP registers while the counter is running. The new value of the STH and STL registers will be loaded at the next End of Count condition, while the new value of the STP register will be loaded immediately.

WARNING: In order to prevent incorrect counting of the Standard Timer, the prescaler (STP) and counter (STL, STH) registers must be initialised before the starting of the timer. If this is not done, counting will start with the reset values (STH=FFh, STL=FFh, STP=FFh).

Single/Continuous Mode.
The S-C bit (STC.6) selects between the Single or Continuous mode.

SINGLE MODE: at the End of Count, the Standard Timer stops, reloads the constant and resets the Start/Stop bit (the user programmer can inspect the timer current status by reading this bit). Setting the Start/Stop bit will restart the counter.

CONTINUOUS MODE: At the End of the Count, the counter automatically reloads the constant and restarts. It is only stopped by resetting the Start/Stop bit.

The S-C bit can be written either with the timer stopped or running. It is possible to toggle the S-C bit and start the Standard Timer with the same instruction.

7.2.2.2 Standard Timer Input Modes (ST9 devices with Standard Timer Input STIN)

Bits INMD2, INMD1 and INEN are used to select the input modes. The Input Enable (INEN) bit enables the input mode selected by the INMD2 and INMD1 bits. If the input is disabled (INEN="0"), the values of INMD2 and INMD1 are not taken into account. In this case, this unit acts as a 16-bit timer (plus prescaler) directly driven by INTCLK/4 and transitions on the input pin have no effect.

Event Counter Mode (INMD1 = "0", INMD2 = "0")

The Standard Timer is driven by the signal applied to the input pin (STIN) which acts as an external clock. The unit works therefore as an event counter. The event is a high to low transition on STIN. Spacing between trailing edges should be at least the period of INTCLK multiplied by 8 (i.e. the maximum Standard Timer input frequency is 3 MHz with INTCLK = 24MHz).

Gated Input Mode (INMD1 = "0", INMD2 = "1")

The Timer uses the internal clock (INTCLK divided by 4) and starts and stops the Timer according to the state of STIN pin. When the status of the STIN is High the Standard Timer count operation proceeds, and when Low, counting is stopped.

Triggerable Input Mode (INMD1 = "1", INMD2 = "0")

The Standard Timer is started by:

a) setting the Start-Stop bit, AND

b) a High to Low (low trigger) transition on STIN.

In order to stop the Standard Timer in this mode, it is only necessary to reset the Start-Stop bit.

Retriggerable Input Mode (INMD1 = "1", INMD2 = "1")

In this mode, when the Standard Timer is running (with internal clock), a High to Low transition on STIN causes the counting to start from the last constant loaded into the STL/STH and STP registers. When the Standard Timer is stopped (ST-SP bit equal to zero), a High to Low transition on STIN has no effect.

7.2.2.3 Time Base Generator (ST9 devices without Standard Timer Input STIN)

For devices where STIN is replaced by a connection to CLOCK2, the condition (INMD1 = "0", INMD2 = "0") will allow the Standard Timer to generate a stable time base independent from the PLL programming.

STANDARD TIMER (Cont'd)

7.2.2.4 Standard Timer Output Modes

OUTPUT modes are selected using 2 bits of the STC register: OUTMD1 and OUTMD2.

No Output Mode (OUTMD1 = "0", OUTMD2 = "0")

The output is disabled and the corresponding pin is set high, in order to allow other alternate functions to use the I/O pin.

Square Wave Output Mode (OUTMD1 = "0", OUTMD2 = "1")

The Standard Timer toggles the state of the STOUT pin on every End Of Count condition. With INTCLK = 24MHz, this allows generation of a square wave with a period ranging from 333ns to 5.59 seconds.

PWM Output Mode (OUTMD1 = "1")

The value of the OUTMD2 bit is transferred to the STOUT output pin at the End Of Count. This allows the user to generate PWM signals, by modifying the status of OUTMD2 between End of Count events, based on software counters decremented on the Standard Timer interrupt.

7.2.3 Interrupt Selection

The Standard Timer may generate an interrupt request at every End of Count.

Bit 2 of the STC register (INTS) selects the interrupt source between the Standard Timer interrupt and the external interrupt pin. Thus the Standard Timer Interrupt uses the interrupt channel and takes the priority and vector of the external interrupt channel.

If INTS is set to "1", the Standard Timer interrupt is disabled; otherwise, an interrupt request is generated at every End of Count.

Note: When enabling or disabling the Standard Timer Interrupt (writing INTS in the STC register) an edge may be generated on the interrupt channel, causing an unwanted interrupt.

To avoid this spurious interrupt request, the INTS bit should be accessed only when the interrupt logic is disabled (i.e. after the DI instruction). It is also necessary to clear any possible interrupt pending requests on the corresponding external interrupt channel before enabling it. A delay instruction (i.e. a NOP instruction) must be inserted between the reset of the interrupt pending bit and the INTS write instruction.

7.2.4 Register Mappingl

Each Standard Timer has 4 registers mapped into Page 11 in Group F of the Register File

In the register description on the following page, register addresses refer to STIM0 only.

STD Timer	Register	Register Address
STIM0	STH0	R240 (F0h)
	STL0	R241 (F1h)
	STP0	R242 (F2h)
	STC0	R243 (F3h)
STIM1	STH2	R248 (F8h)
	STL2	R249 (F9h)
	STP2	R250 (FAh)
	STC2	R251 (FBh)

STANDARD TIMER (Cont'd)

7.2.5 Register Description

COUNTER HIGH BYTE REGISTER (STH)
R240 - Read/Write
Register Page: 11
Reset value: 1111 1111 (FFh)

7							0
ST.15	ST.14	ST.13	ST.12	ST.11	ST.10	ST.9	ST.8

Bits 7:0 = **ST.[15:8]**: *Counter High-Byte.*

COUNTER LOW BYTE REGISTER (STL)
R241 - Read/Write
Register Page: 11
Reset value: 1111 1111 (FFh)

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

Bits 7:0 = **ST.[7:0]**: *Counter Low Byte.*

Writing to the STH and STL registers allows the user to enter the Standard Timer constant, while reading it provides the counter's current value. Thus it is possible to read the counter on-the-fly.

STANDARD TIMER PRESCALER REGISTER (STP)
R242 - Read/Write
Register Page: 11
Reset value: 1111 1111 (FFh)

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

Bits 7:0 = **STP.[7:0]**: *Prescaler.*
The Prescaler value for the Standard Timer is programmed into this register. When reading the STP register, the returned value corresponds to the programmed data instead of the current data.
00h: No prescaler
01h: Divide by 2
FFh: Divide by 256

STANDARD TIMER CONTROL REGISTER (STC)
R243 - Read/Write
Register Page: 11
Reset value: 0001 0100 (14h)

7							0
ST-SP	S-C	INMD1	INMD2	INEN	INTS	OUTMD1	OUTMD2

Bit 7 = **ST-SP**: *Start-Stop Bit.*
This bit is set and cleared by software.
0: Stop counting
1: Start counting

Bit 6 = **S-C**: *Single-Continuous Mode Select.*
This bit is set and cleared by software.
0: Continuous Mode
1: Single Mode

Bits 5:4 = **INMD[1:2]**: *Input Mode Selection.*
These bits select the Input functions as shown in Section 7.2.2.2, Standard Timer Input Modes (ST9 devices with Standard Timer Input STIN), when enabled by INEN.

INMD1	INMD2	Mode
0	**0**	Event Counter mode
0	**1**	Gated input mode
1	**0**	Triggerable mode
1	**1**	Retriggerable mode

Bit 3 = **INEN**: *Input Enable.*
This bit is set and cleared by software. If neither the STIN pin nor the CLOCK2 line are present, INEN must be 0.
0: Input section disabled
1: Input section enabled

Bit 2 = **INTS**: *Interrupt Selection.*
0: Standard Timer interrupt enabled
1: Standard Timer interrupt is disabled and the external interrupt pin is enabled.

Bits 1:0 = **OUTMD[1:2]**: Output Mode Selection.
These bits select the output functions as described in Section 7.2.2.4, Standard Timer Output Modes.

OUTMD1	OUTMD2	Mode
0	**0**	No output mode
0	**1**	Square wave output mode
1	**x**	PWM output mode

7.3 TELETEXT DISPLAY STORAGE RAM INTERFACE

7.3.1 Introduction

The Teletext Display RAM (TDSRAM) is used to hold the Teletext data for display.

It can be shared by the following units:

– Acquisition Unit (AQD). A buffer containing the teletext data extracted by the slicer from the incoming Composite Video signal CVBS1.

– Display Unit (DIS). This OSD generator is described in a separate chapter.

– 40-byte buffer unit (MBT). A register mapped buffer that can be directly accessed by the CPU.

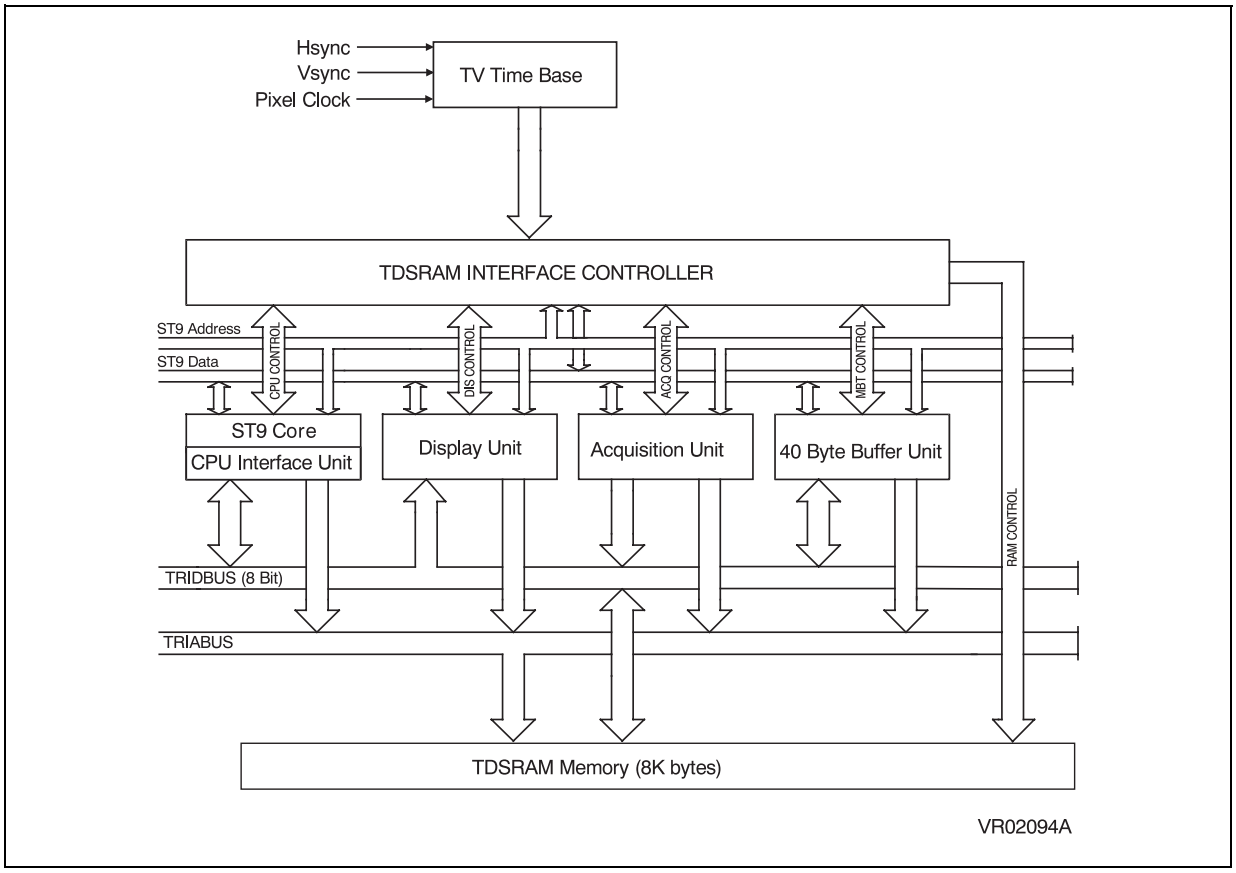
– CPU accesses for control.

The necessary time slots are provided to each unit for realtime response.

FEATURES:

- Memory mapped in CPU Memory Space
- Direct CPU access without significant slowdown
- 3 types of on-chip hardware DMA
- Row-wise DMA for high speed data treatment (bufferized 40-byte read - write)

Figure 47. General Block Diagram



TDSRAM (Cont'd)

7.3.2 Functional Description

The Teletext Data Storage RAM Interface (TRI) manages the data flows between the different sub-units (display, acquisition, 40-byte buffer, CPU interface) and the internal RAM. A specific set of buses (8 bit data TRIDbus, 13 bit address TRIAbus) is dedicated to these data flows.

As this TDSRAM interface has to manage TV oriented real time signals (On-Screen-Display, Teletext slicing storage):

– Its timing generator uses the same frequency generator as for the Display (Pixel frequency multiplier),

– Its controller is hardware synchronized to the basic horizontal and vertical sync signals got through the CSYNC Controller,

– Its architecture gives priority to the TV real time constraints: whenever there is any access contention between the CPU (only in case of direct CPU access) and one of the hardware units, the CPU automatically enters a "wait" configuration until its request is serviced.

7.3.2.1 TV Line Timesharing

During a TV line, to maintain maximum performance, a continuous cycle is run repetitively. This cycle is divided in 8 sub-cycles called "slots".

This 8-slot cycle is repeated continuously until the next TV line-start occurs (horizontal sync pulse detected). When a horizontal sync pulse is detected, the running slot is completed and the current cycle is broken.

The following naming convention is used: "ACQ" stands for Acquisition storage slot, "CPU" stands for direct CPU access slot, "DIS" stands for Display reading slot, "MBT" stands for multi-byte transfer. Each slot represents a single byte exchange (read or write) between the TDSRAM memory and the other units:

Acquisition Storage (ACQ). 1 byte issued from the Teletext Acquisition unit written to the TDSRAM, the address is defined by the acquisition address generator.

Display Reading (DIS). 1 byte is read from the TDSRAM and sent to the display unit, the address being defined by the display address generator.

Multi-Byte Transfer (MBT). 1 byte of the 40 bytes Buffer is exchanged (read or written) between the 40-byte Buffer and the TDSRAM, the address being defined by the 40-byte buffer address generator.

CPU Access (CPU). 1 byte is exchanged (read or written) between the TDSRAM and the CPU, the address being defined by the CPU address bus.

7.3.2.2 TV Field Timesharing

The choice between Acquisition and Display cycles is done automatically on a TV line basis.

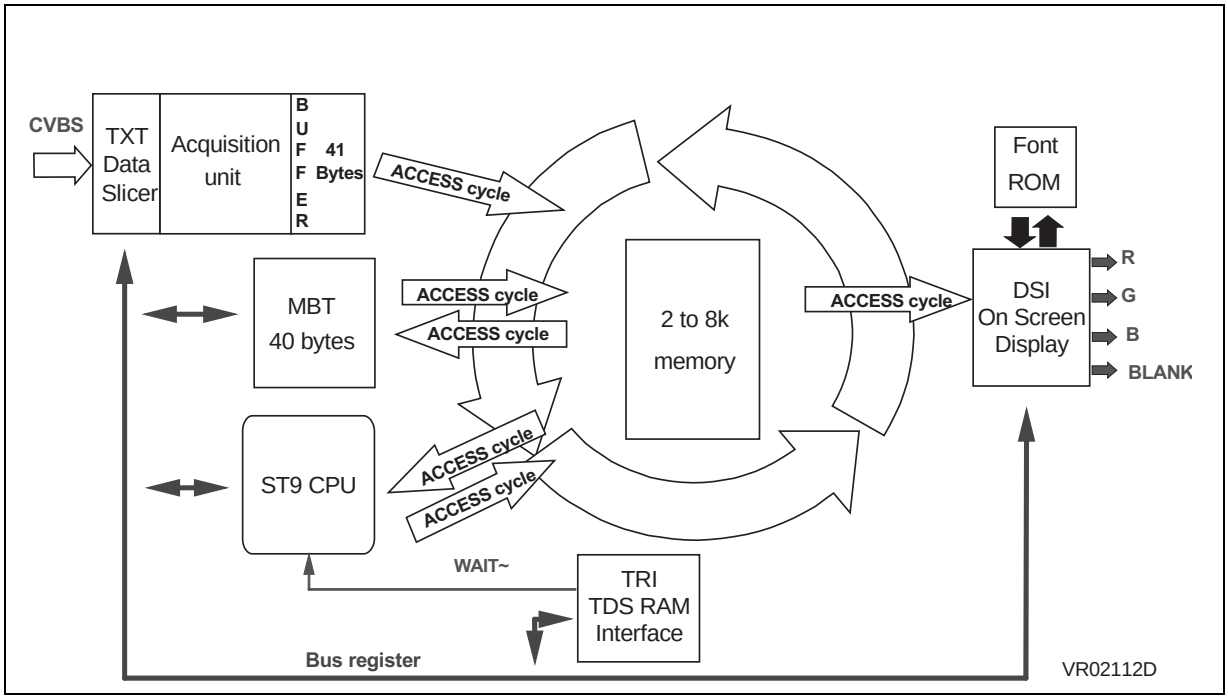
The complete TV field start (VBI) is affected to Acquisition cycles up to end of line 24. The rest of the TV field is affected to Display cycles up to the next field (next vertical sync pulse).

TDSRAM (Cont'd)

Figure 48. Timesharing Slot Configurations

Deflection line 1

Cycle									
	DIS	CPU	MBT	ACQ	DIS	CPU	MBT	CPU	DON = 1 or AON =1, BUSY = 1
	DIS	CPU	CPU	ACQ	DIS	CPU	CPU	CPU	DON = 1 or AON =1, BUSY = 0
	CPU	CPU	MBT	ACQ	CPU	CPU	MBT	CPU	DON = 0 or AON =1, BUSY = 1
	CPU	CPU	CPU	ACQ	CPU	CPU	CPU	CPU	DON = 0 or AON =1, BUSY = 0



TDSRAM (Cont'd)

7.3.2.3 CPU Slowdown on TDSRAM access

As described above, the TDSRAM interface puts priority on TV real time constraints and may slowdown the CPU by inserting wait cycles when a TDSRAM access is requested. The effective duration of the CPU slowdown is a complex function of TDSRAM interface working mode and of the respective DOTCK/2 frequency (TDSRAM frequency) and the Core INTCLK frequency.

In order to calculate the average and worst case slowdown, let's define the following parameters:

INT(): stands for "integer" function

TCPU: CPU internal clock period

TDRAM: TDSRAM clock period (DOTCLK/2 period)

TWAIT: additional time inserted due to the TDSRAM access

S: number of elapsed slots to get a CPU slot (this may be a real number)

TWAIT (read) = INT(2.5 * S * (TDRAM/ TCPU)) * TCPU + (+1 / -0) * TCPU

TWAIT (write) = INT(2.5 * S * (TDRAM/ TCPU) + 1) * TCPU + (+1 / -0) * TCPU

DIS (or ACQ) slot	MBT slot	Average S	Max S
off	on	12/8 = 1.5	3
off	off	9/8 = 1.16	2
on	off	12/8 = 1.5	3
on	on	16/8 = 2	4

Assuming the Display is "on", no "MBT" is required and we have the following clock conditions:

CPU running at 12 MHz (TCPU= 83ns)

DOTCK/2 at 20 MHz; 4/3 recommended frequency (TDRAM= 50ns) the average number of inserted wait cycles is:

TWAIT(read) = 2 * TCPU + (+1 / -0) * TCPU
(i.e. 2 or 3 extra CPU cycles)

TWAIT(write) = 3 * TCPU + (+1 / -0) * TCPU
(i.e. 3 or 4 extra CPU cycles)

In practice:

ld (rr), #N will last 11 or 12 cycles instead of 8 cycles

ldw (rr), #NN will last 20 to 22 cycles instead of 14 cycles

TDSRAM (Cont'd)

7.3.3 Initialisation

7.3.3.1 Clock Initialisation

Before initialising the TRI, first initialise the pixel clock. Refer to the Application Examples in the OSD chapter and to the RCCU chapter for a description of the clock control registers.

7.3.3.2 TRI Initialisation

It is recommended to wait for a stable clock issued from the Pixel frequency multiplier before enabling the TDSRAM interface.

Use the CONFIG register to initialise and start the TRI and Acquisition units. Note: The AON and DON bits can only be changed while GEN=0

Example:

```
spp #0x26
ld config, #0x06 ; AON,DON,GEN=0
or config, #0x01 ; set GEN=1
```

During and after a reset, the TDSRAM interface is forced into its "disable" mode where the sequencer is forced into its idle state.

7.3.3.3 Multi-Byte Transfer (MBT) Initialisation

A multi-byte transfer corresponds to a 40-byte exchange between the RAM and an internal 40-byte buffer located into the TDSRAM interface. This buffer is register-mapped and can be directly accessed by the CPU in its register page space.

Start the MBT transfer by setting the BUSY bit in the BUFC register.

Example:

```
spp #0x26
ld BUFC, #0x01 ;
;Start DMA transfer
;Poll on Busy bit
```

The exchange can be either a read (extraction of 40 consecutive bytes from the RAM starting at a software programmed address) or a write (writing of 40 consecutive bytes to the RAM starting at a software programmed address). The address is a 13 bit-long word allowing access to any TDSRAM location. The address can be either incremented or decremented depending on a control bit.

While the transfer is running, the buffer is no longer software accessible ("busy bit" (BUFC.0) is set to 1). Once the exchange is completed, this bit is automatically reset and the MBT slots are automatically given back to the CPU.

Four powerful data exchange modes are provided:

– Read only (transfer from the RAM to the buffer)

– Write only (transfer from the buffer to the RAM)

– Write with parity reject (if the byte presents a parity error, it will not be written into the TDSRAM; the corresponding location keeps its previous content).

– Parity cancelled on write (the MSB is replaced by "0" when the byte is written into the TDSRAM).

The "parity reject" and "parity cancelled" modes can be used simultaneously during a write operation. In this case, the parity check will be done first and the parity bit will be removed, if the write operation has to be performed. The parity check performed is the following:

– The parity is correct when the number of 1s (current byte) is an odd number.

– The parity is incorrect when the number of 1s (current byte) is an even number.

A parity check flag (BUFC.5) is provided for the whole buffer. This bit is set when a parity error is detected during the write operation. This bit has to be reset by software before starting another MBT.

7.3.3.4 100/120 Hz Applications

In 100/120 Hz applications, both the vertical and horizontal beam scanning speeds are doubled while the CVBS signal remains unchanged. To handle this, the EOFVBI interrupt can be delayed from the beginning of deflection line 25 to the beginning of line 50 by setting the DS bit of the CONFIG register. If the DS bit is set, the interrupt is only generated when the complete Teletext data is fully sliced and stored.

TDSRAM (Cont'd)

7.3.4 Register Description

7.3.4.1 Data Registers

BUF0..15 R240 .. R255 Page 36Read/Write

RAM Buffer Data Register x = 0,..,15

BUF16..31 R240 .. R255 Page 37 Read/Write

RAM Buffer Data Register x = 16,..,31

BUF32..39 R240 .. R247 Page 38 Read/Write

RAM Buffer Data Register x = 32,..,39

Reset Value: xxxx xxxx (xxh)

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

Bits 7:0 = **BUFx[7:0]**: presents the x-th byte (MSB...LSB) of the 40 byte data buffer belonging to the teletext function. The data registers are not accessible during the transfer (BUSY = "1"). A buffer exchange starts with BUF0 and ends with BUF39 whether the address is incremented or decremented.

7.3.4.2 Address Registers

MULTI-BYTE TRANSFER START ADDRESS REGISTER 1 (MBTSA1)
R250 - Read/Write
Register Page: 38
Reset Value: 1000 0000 (80h)

7							0
1	0	SA13	SA12	SA11	SA10	SA9	SA8

Bits 7:6 = Hard wired to map correctly into the memory space, defined for the RAM.

Bit 5 = **SA[13]**: This bit must always be written to 0.

Bits 4:0 = **SA[12:8]**: Bit13..bit8 of the 13-bit start address for the 40-byte data transfer have to be written into this register.

MULTI-BYTE TRANSFER START ADDRESS REGISTER 0 (MBTSA0)
R251 - Read/Write
Register Page: 38
Reset Value: 0000 0000 (00h)

7							0
SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

Bits 7:0 = **SA[7:0]**: Bit7..bit0 of the 13-bit start address for the 40-byte data transfer have to be written into this register.
MBTSA1..MBTSA0 are not accessible during the transfer (BUSY = "1").
MBTSA is used as a counter to generate the TDSRAM (R/W) address. When the Multi-Byte Transfer is done, this register is incremented or decremented by 40 depending on the BADU bit in the BUFC register R248 page 26h.
The normal mode corresponds to a counter incrementation.
When the user wants to use decrementation mode (using the BADU bit), the address must be complemented before writing to the 13 LSB of MBTSA1.. MBTSA0.

For example, to decrement the counter from the address:

– SA12 SA11..... SA0 = 1 0101 0101 0101b = 1555h,

the software must load:

– SA12n SA11n..... SA0n = 0 1010 1010 1010b = 0AAAh,

in the start address register. The BADU bit must be set to 1.

The value read in this case will be MBTSA1.. MBTSA0 = 8AAAh (The 3 MSB are hard wired).

TDSRAM (Cont'd)

7.3.4.3 Control Registers

RAM BUFFER CONTROL REGISTER (BUFC)
R248 - Read/Write
Register Page: 38
Reset Value: 0000 1000 (08h)

7							0
0	0	PEF	BADU	MOD2	MOD1	MOD0	BUSY

Bits 7:6 = Reserved, keep in reset state.

Bit 5 = **PEF**: *Parity Error Flag*.

This bit is set by hardware, when a parity error has occurred during the 40 byte transfer (in any write mode).
PEF has to be reset by software before starting another MBT.

Bit 4 = **BADU**: *Buffer Address Down/Up*.
This bit is set and cleared by software.
0: Address counter in incrementation mode
1: Address counter in decrementation mode

Bits 3:1 = **MOD[2:0]**: *Multi-Byte Transfer Mode Select Bits*.
Programming these bits, allows the user to choose

MOD2	MOD1	MOD0	Selected Mode
0	0	0	Write only
0	0	1	Write with parity reject
0	1	0	Parity cancelled on write
0	1	1	Write with parity reject & parity cancelled on write
1	-	-	Read only

With the chosen coding, MOD2 serves at the same time as a buffer read/write signal.

The selected mode is memorised when "BUSY" is set. Any further modification of the 3 bits will only be taken into account for the next MBT. The reset value corresponds to a read access of the RAM.

Bit 0 = **BUSY**: *Multi-Byte Transfer Busy Bit* (R/W)
When this bit is set by software, the RAM interface starts a 40 byte transfer. As long as this "busy flag" is set, the RAM interface doesn't accept a new transfer request (buffer is "busy"). BUSY is automatically reset when the transfer has been finished. The RAM access slots, reserved for the MBT, are used for direct CPU access when BUSY is "0".

TDSRAM (Cont'd)

RAM INTERFACE CONFIGURATION REGISTER (CONFIG)
R252 - Read/Write
Register Page: 38
Reset Value: 0000 0110 (06h)

7							0
0	0	0	0	DS	AON	DON	GEN

Bits 7:4 = Reserved, keep in reset state.

Bit 3 = **DS:** *Double Scan*

When the DS bit is reset, the TDSRAM interface and the CSYNC controller behave in 50Hz/60Hz compatible mode. The acquisition storage is only allowed up to the end of line 24. The EOFVBI interrupt is generated at the beginning of line 25.

When the DS bit is set, the TDSRAM interface and the CSYNC controller behave in 100/120Hz compatible mode. The EOFVBI interrupt is generated at the beginning of deflection line 50.

Note: DS can be changed only when the TRI is off (GEN = 0).

Bit 2 = **AON**: *Acquisition ON/OFF.*
0: No acquisition storage allowed (acquisition slot completely used for CPU access).
1: Acquisition storage enabled during the respective access slot.

Note: AON can be changed only when the TRI is off (GEN = 0).

Bit 1 = **DON**: *Display ON/OFF*.
0: No display reading allowed (display slot completely used for CPU access).
1: Display reading enabled during the respective access slot.

Note: DON can be changed only when the TRI is off (GEN = 0).

Bit 0 = **GEN**: *RAM Interface General Enable.*
0: TRI off. Acquisition storage, display reading, multi-byte transfer and CPU accesses are not allowed. When GEN=0, the Automatic Wait Cycle insertion, while trying to access the TDSRAM, is disabled.
1: TRI on.

7.4 ON SCREEN DISPLAY (OSD)

7.4.1 Introduction

The OSD displays Teletext or other character data and menus on a TV screen.

In **serial mode**, characters are coded on one byte. The display is fully compliant with the WST Teletext level 1.5.

In **parallel mode**, characters are coded on two bytes, one byte being the font address (character code), the second byte being used for attribute control, which can be combined with the serial attribute capabilities. In this mode, the display meets a significant part of the WST Teletext level 2 specification.

In order to save memory resources (reduce system cost), two display modes are provided with either a **page mode** display mode (teletext standard, 26 rows) or a **line mode** (up to 12 rows) for non teletext specific menus.

The OSD is seen by the ST9 as a peripheral which has registers mapped in the Paged Register space.

The character codes to be displayed are taken from the TDSRAM memory. They are addressed by the display with the real time sequencer through the TDSRAM interface character by character.

The font ROM contains 2 sets of 512 characters. The standard European font contains all characters required to support Eastern and Western European languages. Each character can be defined by the user with the OSD Screen/Font Editor. All fonts (except the G1 mosaic font) are fully definable by masking the pixel ROM content.

Display is done under control of the ST9 CPU and the vertical and horizontal TV synchro lines.

The OSD provides the Red, Green, Blue signals and the Fast Blanking switching signal through four analog outputs. The three Color outputs use a 3-level DAC which can generate half-intensity colors in addition to the standard saturated colors.

The Display block diagram is shown on Figure 49 on page 96.

A smart pixel processing unit provides enhanced features such as rounding or fringe for a better picture quality. Other smart functions such as true Scrolling and cursor modes allow designing a high quality display application.

7.4.2 General Features

- Serial Character Mode supporting Teletext level 1.5
- Parallel Character Mode for TV character displays (for example channel selection or volume control menus)
- 40 or 80 characters/row
- Full Page Mode:
  23 rows plus 1 Header and 2 Status Rows
- Line Mode:
  Up to 12 rows plus 1 Header and 2 Status Rows.
- 4/3 or 16/9 screen format
- Synchronization to TV deflection, by Hsync and Vsync or Csync.
- Box Mode: Display text inside and outside box solid, transparant or blank
- Rounding and Fringing
- Cursor Control
- Concealing
- Scrolling
- Semi-transparent mode (text windowing inside video picture)
- Half-Tone mode (reduces video intensity inside a box)
- Normal character size 10 x 10 dots.
- Other character sizes available as follows:

Both Serial and Parallel Mode	Parallel Display Mode only
SH x SW = 10 * 10 dots	SH x DW = 10 * 20 dots
DH x SW = 20 * 10 dots	DS=DH x DW = 20 * 20 dots

ON SCREEN DISPLAY (Cont'd)

- Serial character attributes:
  - Foreground Color (8 possibilities in Serial Full page display mode)
  - Background Color (8 possibilities)
  - Flash / Steady
  - Start Box / End Box
  - Double height
  - Conceal / display
  - Fringe
  - Contiguous Mosaic / Separated Mosaic
  - Hold / Release Mosaic
  - G0 font switch (in triple G0 mode)
- Parallel character attributes (in parallel display mode):
  - Underline
  - Double height & Double width
  - Upper Half-Character
  - Smooth Rounding
  - Box mode
  - Font Selection G0/Extended menu
  - Selection of 15 background Colors
  - Selection of 8 foreground Colors
- Global Screen attributes:
  - Fine and coarse Horizontal Adjustment (for the whole 26 rows)
  - Vertical Adjustment (for the whole page)
  - Blanking Adjustment
  - Default Background Color (up 15 colors with use of half-intensity attribute)
  - Default Foreground Color (up 15 colors with use of half-intensity attribute)
  - Semi transparent display (active only on background)
  - Translucency: OSD background color mixed with video picture.
  - Full screen Color (15)
  - National Character set selection
  - National Character mode selection
  - Global Double Height display (Zooming Function)
  - Global Fringe Enable
  - Global Rounding Enable
- Cursor Control:
  - Horizontal position (by character)
  - Vertical position (by row)
  - Flash, Steady or Underline Cursor Modes
  - Color Cursor with inverted foreground / inverted background
- Scrolling Control:
  - Vertical scrolling available: Programmable rolling window if Normal Height and 40 char/row
  - Top-Down or Bottom-Up shift
  - Freeze Display
- Character fonts:

  1088 different characters available:

  – 128 mosaic matrix characters (G1), hardware defined (64 contiguous, 64 separated).
  - 2 x 512 character ROM fonts, all user defined:
  - 96-character basic character set (G0)
  - 128 characters shared between G2 X/26 and Menu characters
  - 96 Extended Menu Characters
  - Two national character set modes (mutually exclusive ROM options):

    Single G0 mode

    A font combining 83 characters from the G0 basic set (latin) and 13 characters selected from 15 National character subsets

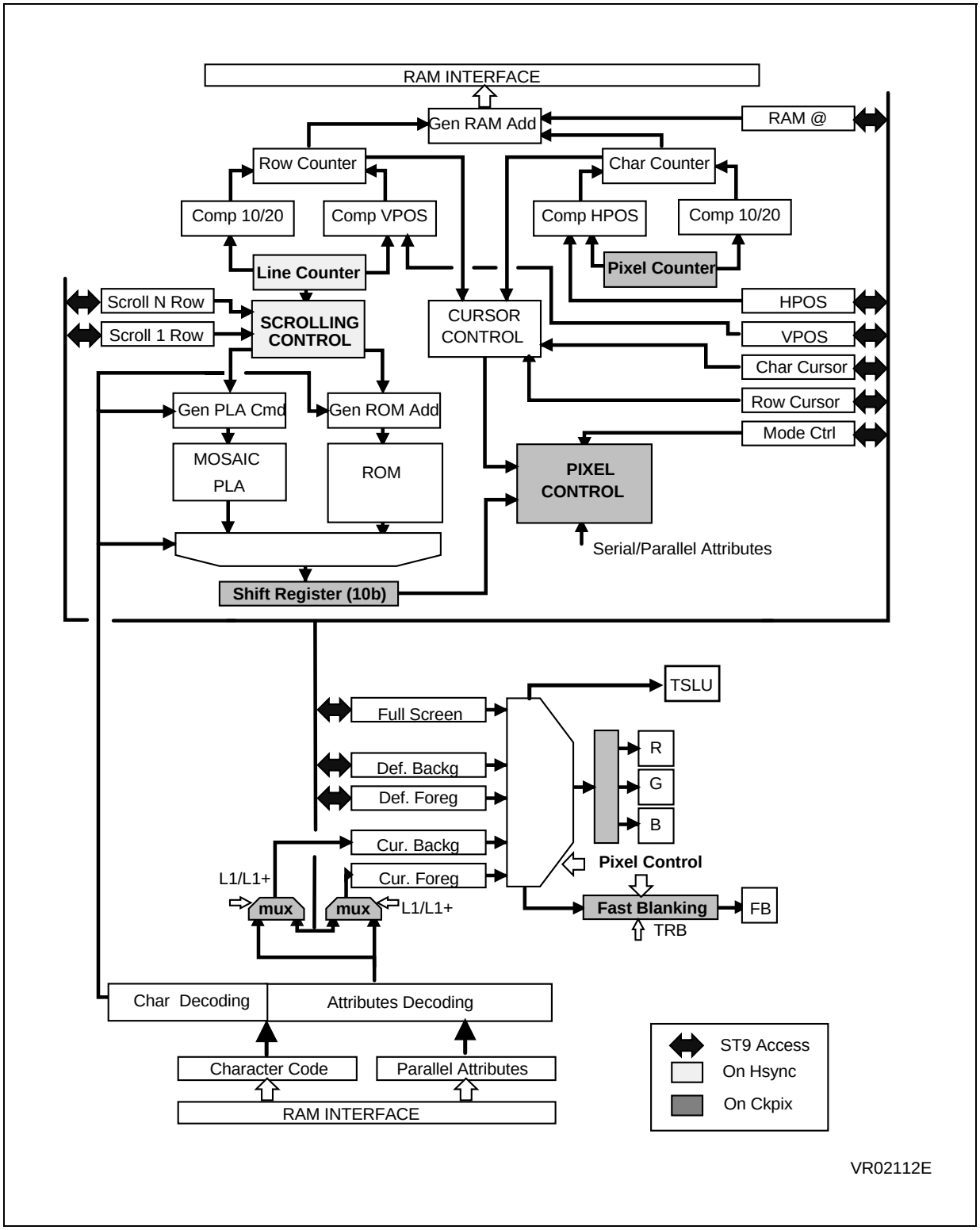
    Triple G0 mode allowing different alphabets

    Three 96-character fonts (e.g. latin, arabic, cyrillic ...)

Mode	G0	National Set	G2 (X26+ Menu)	Extended Menu	G1 (mosaic)
Triple G0	3*96	N/A	128	96	64
Single G0	1*83	15*13	128	96	64

ON SCREEN DISPLAY (Cont'd)

Figure 49. Display Block Diagram



ON SCREEN DISPLAY (Cont'd)

7.4.3 Functional Description

7.4.3.1 Screen Display Area

The screen is divided in 26 rows of basically 40 characters. From row 1 to row 23, it is possible to display 80 characters per row with the following restrictions:

– Serial mode only

– No rounding or fringe

The three special rows, a Header and two Status rows have specific meanings and behaviour. They are always displayed the same way (40 characters) and at the same place. In these rows, size attributes, scrolling and 80-character modes are not allowed.

All row content, including the Header and Status rows, is fully user-definable.

Figure 50. Definition of Displayed Areas

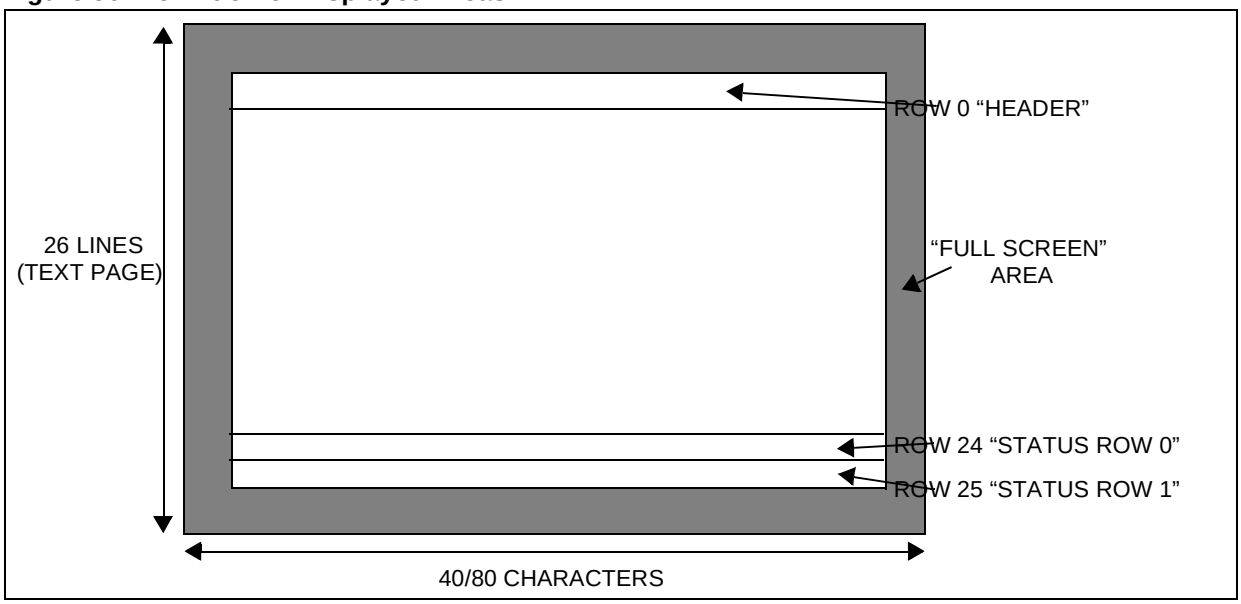


Figure 51. Screen Display Area.



ON SCREEN DISPLAY (Cont'd)

7.4.3.2 Color Processing

The color of any pixel on screen is the result of a priority processing among several layers which are (going from the lowest priority to the highest one):

- Full Screen Color where nothing is processed
- Default Background Color (it assumes pixel is off)
- Serial Background Color (pixel off, but background color serial attributes activated)
- Parallel Background Color (pixel off, but background color parallel attribute activated)
- Default Foreground Color (pixel on, but no foreground attribute activated)
- Serial Foreground Color (pixel on and foreground serial attribute activated)
- Parallel Foreground Color (pixel on and foreground parallel attribute activated)

Color processing is also the result of register control bits (for global color attributes) and color oriented attribute bits (from serial or parallel attributes), refer to the 7.4.4.3 on page 107

7.4.3.3 Pixel Clock Control

The pixel clock is generated outside of the display macrocell by the on-chip Pixel Frequency multiplier which provides great frequency flexibility controlled by software (refer to the RCCU chapter). For example, reconfiguring the application from a 4/3 screen format to a 16/9 format is just a matter of increasing the pixel frequency (i.e. reprogramming the pixel frequency multiplier to its new value).

The output signal of the pixel frequency multiplier is rephased by the Skew Corrector to be perfectly in phase with the horizontal sync signal which drives the display.

7.4.3.4 Display Character

Each character is made up of a 10 x 10 dots matrix. All character matrix contents are fully user definable and are stored in the pixel ROM (except the G1 mosaic set which is hardware defined).

A set of colors defines the final color of the current pixel.

In general, the character matrix content is displayed as it is, the pixel processing adding the shape and the color information received from the current attributes. Only three kinds of attributes alter the displayed pixel. They are the following:

7.4.3.5 Rounding

Rounding can be enabled for the whole display using the GRE global attribute bit (See Table 18. on page 100) In this effect one half-dot is added in order to smooth the diagonal lines. This processing is built into the hardware. The half-dot is painted as foreground. This half-dot is field-sensitive for minimum vertical size (Figure 52 on page 99).
An extra 'smooth rounding' capability is also built-in (see Figure 53 on page 99). In smooth rounding, a pixel is added even if dots make an 'L'. This capability is activated using a parallel attribute (See Table 21 Parallel Color and Shape Attributes.)

7.4.3.6 Underline

In this effect the last TV line of the character is displayed as foreground (Figure 52 on page 99).

7.4.3.7 Fringe

The fringe is a half-dot black border surrounding completely the character foreground. This half-dot is field sensitive for minimum vertical size (Figure 52 on page 99).

7.4.3.8 Translucency

Certain video processors are able to mix the RGB and video signals. This function of the chroma processor is then driven by the TSLU output pin of the ST9 device. See Figure 55 on page 102.

7.4.3.9 Half-Tone

If the HT signal is activated, for example, while a text box is displayed and a transparant background selected for all the display (MM bit =1 in the FSCCR register), the HT signal performs a contrast reduction to the background inside the box. See Figure 56 on page 103.

ON SCREEN DISPLAY (Cont'd)

Figure 52. Display Character Scheme

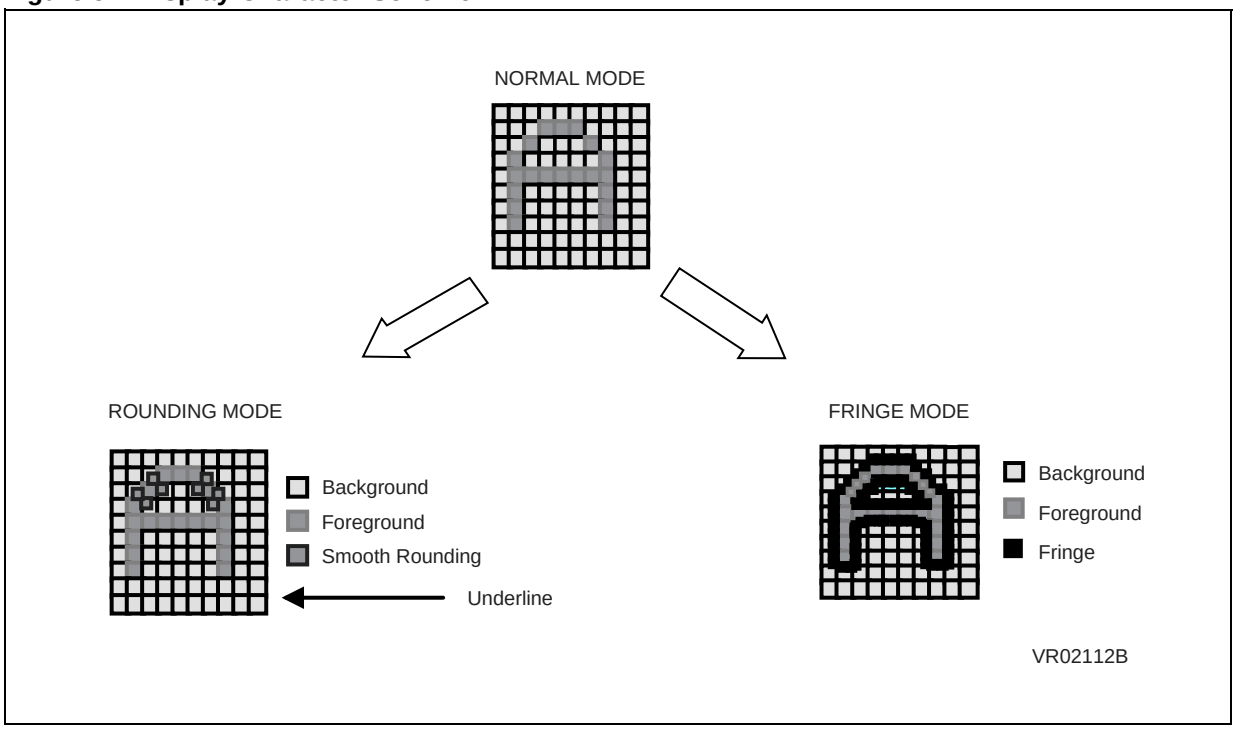
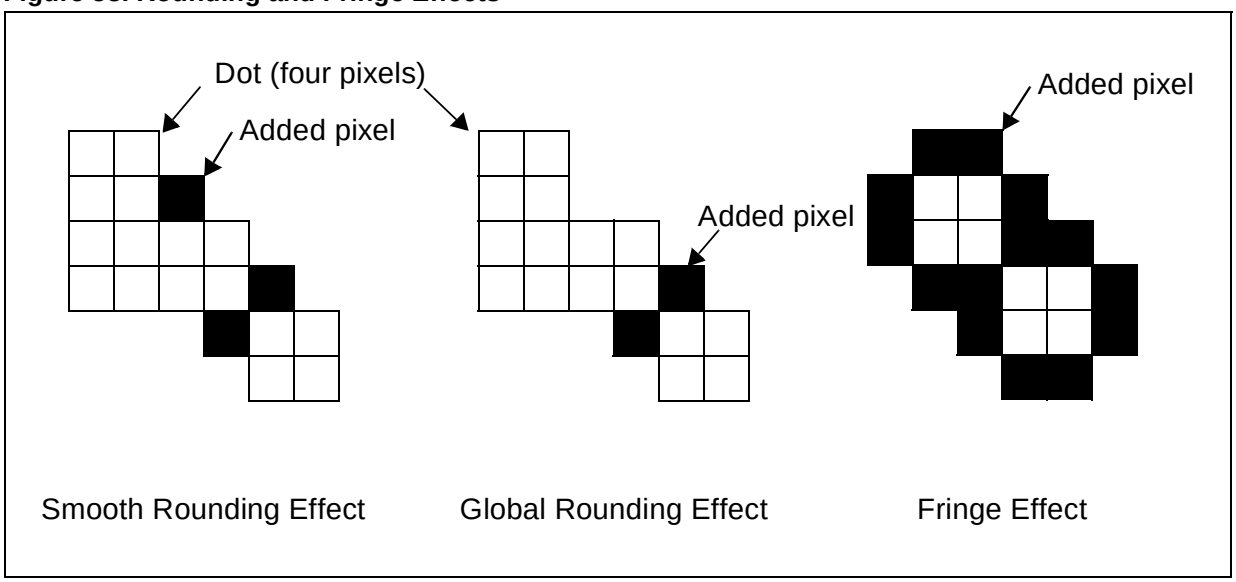


Figure 53. Rounding and Fringe Effects



ON SCREEN DISPLAY (Cont'd)

7.4.4 Programming the Display

All the characteristics of the display are managed by programmable attributes:

- Global Attributes
- Serial Attributes
- Parallel Attributes (active until a superseding serial or parallel attribute).
- Cursor Control
- Scrolling Control

7.4.4.1 Global Attributes

These global attributes are defined through their corresponding registers (see the Register Description).

Table 18. Global Attributes

Global Attributes	Description	Control Register
Display Enable (DE)	0= Display Off (Default) 1= Display On	DCM0R R250 (FAh) Page 32
4/3 or 16/9 Format (SF)	0= 4/3 Screen Format (Default) 1= 16/9 Screen Format	DCM0R
Conceal Enable (CE)	0= Reveal any text defined as concealed by serial attributes (Default) 1= Conceal any text defined as concealed by serial attributes	DCM0R
Fringe Enable (FRE)	0= Fringe Disabled (Default) 1= If SWE in NCSR register is reset, it acts as Fringe enable (toggle with serial attribute 1Bh). Active on the whole page but not in 80-character mode.	DCM0R
Global Fringe Enable (GFR)	0= Global fringe mode off 1= Display all text in page in fringe mode	DCM0R
Global Rounding Enable (GRE)	0= Disabled (Default) 1= Rounding active on the whole page but not in 80-character mode.	DCM0R
Semi-transparent Mode (STE)	0=Disabled (default) 1=Enabled The Fast Blanking signal is toggled with the double pixel clock rate on Background and full screen area in 40 character mode. Note: Semi-transparent mode shows a visible grid on screen.	DCM0R
Translucency (HTC and TSLE)	The TSLU signal is active when the OSD displays the background and full screen area and is inactive during foreground or if no display. This output pin is used with a Chroma processor to mix the video input with the RGB to get full translucency.	NCSR R245 (F5h) Page 32 and FSCCR R243 Page 32
Half-Tone (HTC and TSLE)	The HT signal is active when the OSD displays the background and full screen area and is inactive during foreground or if no display. The HT signal is used with a video processor to perform a contrast reduction.	NCSR R245 (F5h) Page 32 and FSCCR R243 Page 32
40/80 Chars/Row (S/D)	0=Single page (40 Characters per row) (default) 1= Two pages are displayed contiguously (80 Characters per row). In this mode, only serial mode is available.	DCM0R
Fast Blanking Active Level	0=Display when Fast Blanking output is low (default) 1=Display when Fast Blanking output is high	DCM1R R251 (FBh) Page 32
Serial/Parallel Mode (SPM)	0= Serial Mode (Default) 1= Parallel Mode	DCM1R
Page or Line Display Mode (PM)	0 = Full Page Mode (Default) 23 lines plus 1 header and two status lines. 1= Line Mode	DCM1R

ON SCREEN DISPLAY (Cont'd)

Table 19. Global Attributes (Cont'd)

Global Attributes	Description	Control Register
Box Control ModeText In/ Out	Text In/ Text Out Box configurable with 3 bits. Refer to FSCCR register description for details.	FSCCR R243 (F3h) Page 32
Vertical Adjustment	Refer to the register description for bit settings. Active on the whole page, this setting adjusts the vertical delay between the rising edge of Vsync and the beginning of the display area. The display color in this delay adjustment area is defined by the Full Screen color.	VPOSR R242 (F2h) Page 32
Horizontal Adjustment	Refer to the register description for bit settings. Active on the whole page, this adjustment is the horizontal delay between the rising edge of Hsync and the beginning of the display area. The display in this delay area is the full row color. Two kinds of horizontal adjustment are available. When the tube is in a 4/3 format, only a horizontal delay is necessary before starting the active display area. When the tube is in 16/9 format, an additional horizontal adjustment is necessary to keep the display area centered on the screen.	HPOSR R241 (F1h) Page 32
National Character Subset Selection	Refer to the register description for bit settings. Chooses which national font sub-set is to be used with the G0 character set.	NCSR R245 (F5h) Page 32
Single G0 or Triple G0 mode selection	0 = Single G0 character set mode (default) 1 = Triple G0 character set mode In applications with multiple alphabets in the same display, it is possible to switch from one character set to another on the fly (see serial attributes).	NCSR
Global Double Height	Active on the whole page with header, but not on the status rows. When Global Double Height is active, either the top half or the bottom half of the screen is visible.	SCLR R248 (F8h) and SCHR R249 (F9h) Page 32
Background default color	This color is displayed as background color if no serial or parallel attributes are defined for the displayed row.	DCR R240 (F0h) Page 33
Foreground default color	This color is displayed as foreground color if no serial or parallel attributes are defined for the displayed row. These default colors are selected at each beginning of a line and are defined by means of the corresponding register.	DCR
Full screen color	Color displayed outside of the vertical display area.	FSCCR R243 (F3h) Page 32

ON SCREEN DISPLAY (Cont'd)

Figure 54. Semi-Transparent Display Scheme and Fast Blanking Behaviour

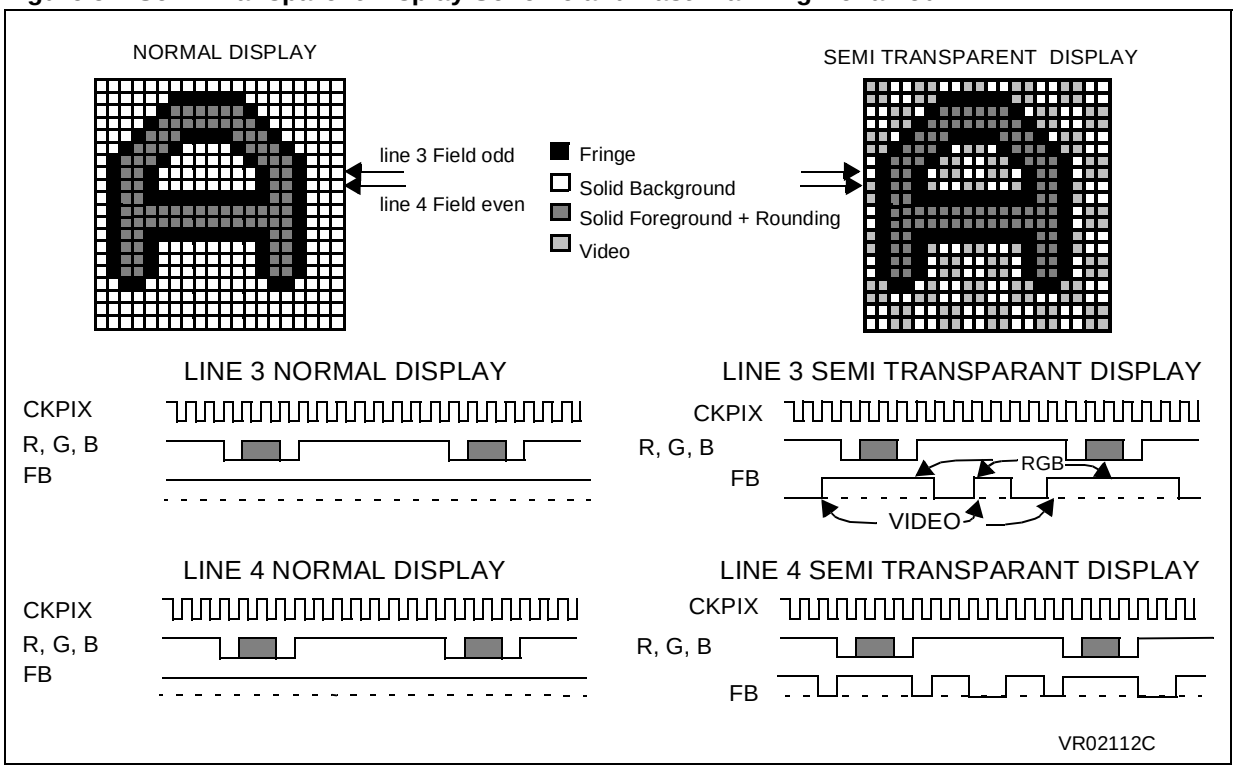
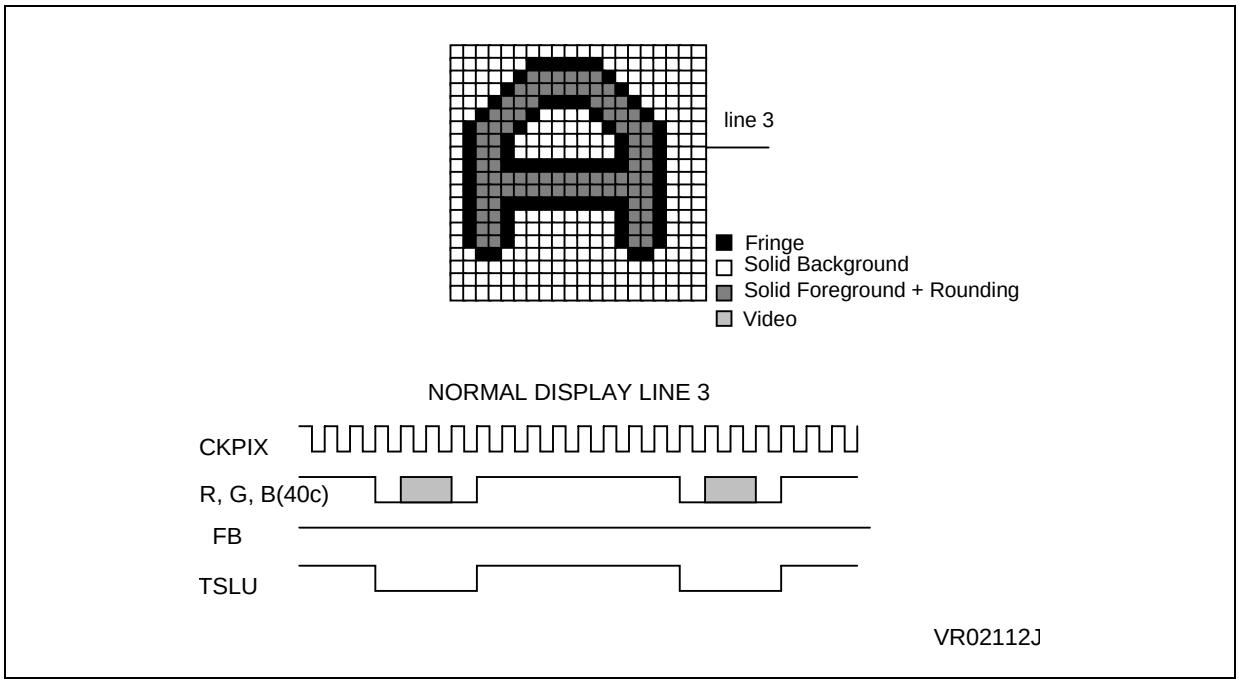
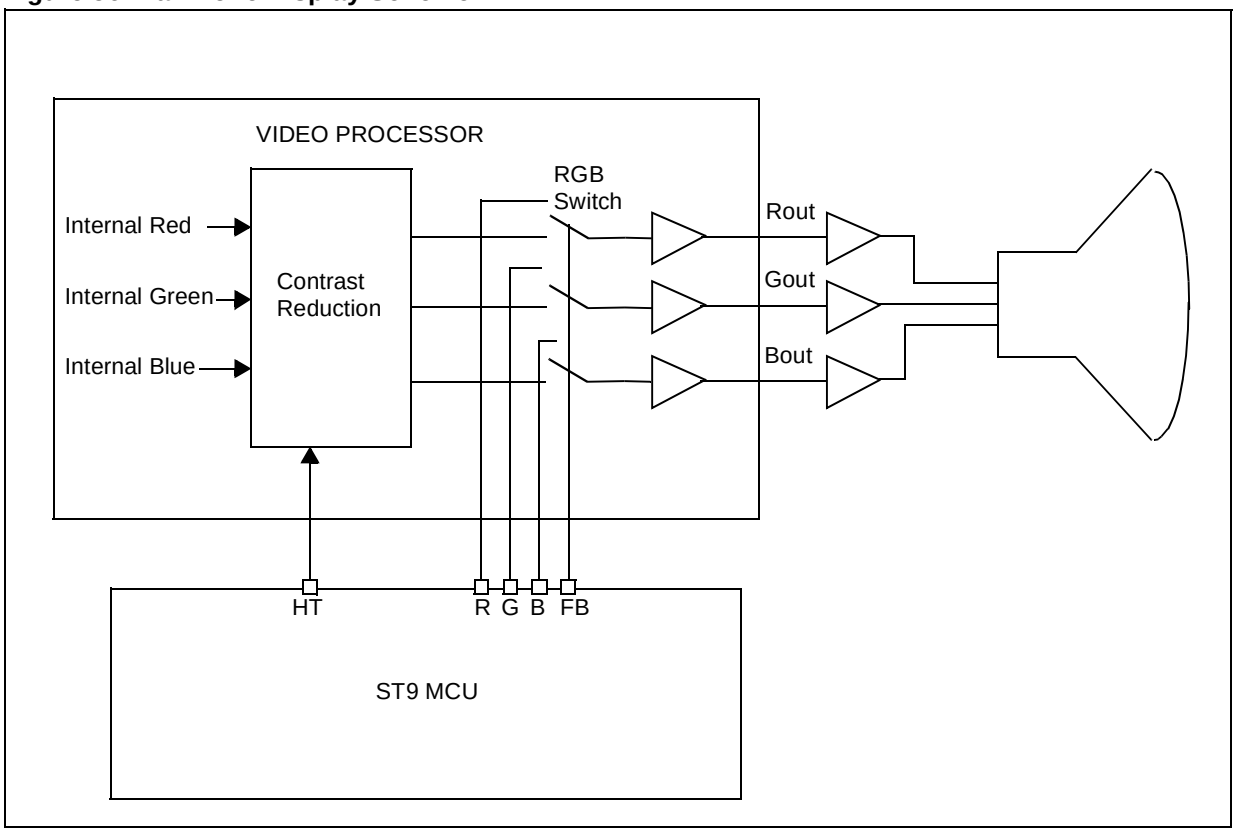


Figure 55. Translucent Display Scheme



ON SCREEN DISPLAY (Cont'd)

Figure 56. Half-Tone Display Scheme



7.4.4.2 Row Attributes

The header and status row attributes are set using the **HSCR** R244 (F4h) Page 32 register. The row enable bits as set in registers **DE0R .. 2** R253 ..255 Page 32.

Header Enable

When the display is in line mode, row 0, called the header, is also usable. It no longer acts as a header but simply as an additional row.

Status Row Enable

The display of the two status rows can be enabled individually.

Row Enable Bits

1 bit per row, for rows from 1 to 23, in page mode.

Serial Attributes

Serial Mode is selected by resetting the **SPM** bit in register **DCM1R** R251 (FBh) Page 32.

Serial attributes are active until the end of the line or a superseding serial attribute.

In this display mode, the attribute code and the character code are in the same memory area (Figure 57 on page 104).

The attribute takes the place of an alpha character, and the OSD displays a space character defined on 1 byte in Serial Mode.

ON SCREEN DISPLAY (Cont'd)

Figure 57. Example of a Row in Serial Mode

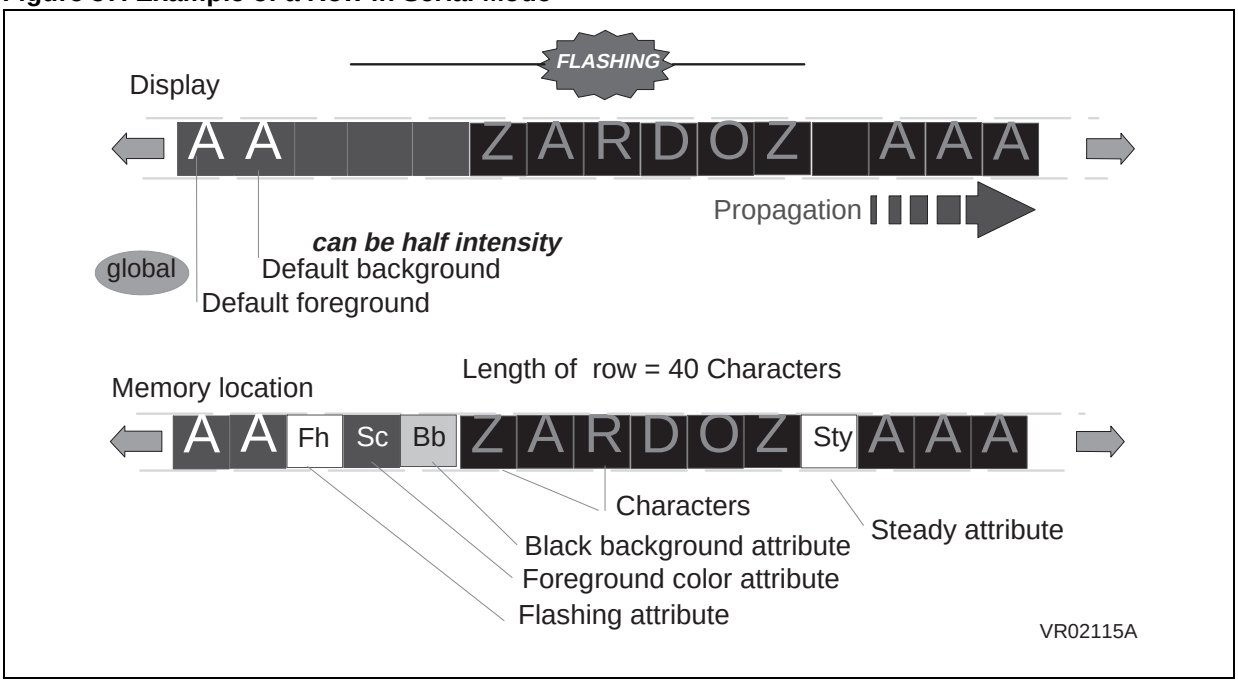


Table 20. Serial Attribute Codes

b[7:3] / b[2:0]	00000	00001	00010	00011
	Foreground Color (Alpha Chars)		**Foreground Color (Mosaic Chars)**	
000	Black	Flash	Black (3)	Conceal (2)
001	Red	Steady (1, 2)	Red	Contiguous Mosaic (1, 2)
010	Green	Box OFF (1)	Green	Separated Mosaic (2)
011	Yellow	Box ON	Yellow	Fringe or 2nd G0 font (3, 4)
100	Blue	Normal Height (1,2)	Blue	Black Background (1, 2)
101	Magenta	Double Height	Magenta	New Background (2)
110	Cyan		Cyan	Hold Mosaic (2)
111	White (1)		White	Release Mosaic (1)

Notes:

(1) Presumed at the start of each display row or can be defined in global register

(2) Action "set at" (on current character) others are "set after" (on next character)

(3) ALWAYS active (even in Full Page Serial Mode, i.e. for Text Level 1)

(4) Toggles action if the Fringe Enable is set (bit 5 in register DCM0R R250 (FAh) Page 32. Selects a second G0 if the Switch Enable bit is set (bit 5 in register NCSR R245 (F5h) Page 32)

ON SCREEN DISPLAY (Cont'd)

Flash: (/= Steady) The next characters are displayed with the foreground color alternatively equal to background and foreground on a period based on Vsync (32 Vsync: foreground, 16 Vsync: background) until a Steady serial attribute.

Fringe: If the Fringe Enable bit is set in the global attribute register DCM0R R250 (FAh) Page 32, the next characters are displayed with a black fringe (half dot) until the decoding of another fringe attribute coded 1Bh (toggle effect).

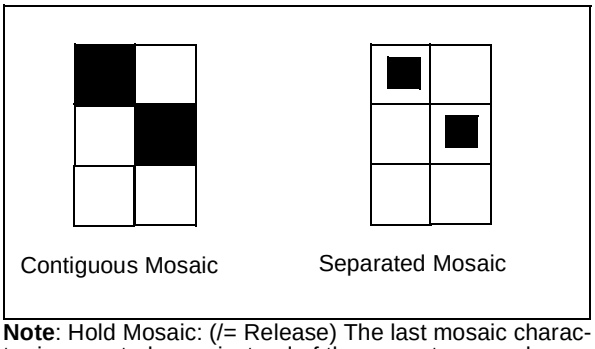
Conceal: (/= Reveal) The next characters are displayed as space characters (Background color) until a foreground color character is encountered. Conceal mode is set by the conceal enable control bit in the register DCM0R R250 (FAh) Page 32.

Boxing: A part of the page (where this bit is active) is inserted in a specific window depending on 3 control bits defined in the FSCCR register. (see Figure 59 on page 106)

To respect the **Teletext Norm**, the box in serial mode, starts when two Box-on attributes are encountered, and stops when two Box-offs are encountered.

Double Height: The upper halves of the characters are displayed in the current row, the corresponding lower halves of characters are displayed (with same display attributes) in the next row (information received for this row must be ignored).
Note: When a serial double height attribute is decoded in Row 23, the characters of the first status row are not displayed. To avoid this effect, remove the serial double height attribute from Row 23.

Figure 58. Mosaic Characters



Note: Hold Mosaic: (/= Release) The last mosaic character is repeated once instead of the current space character.

Figure 59. Example of Boxing Attribute in Serial Mode

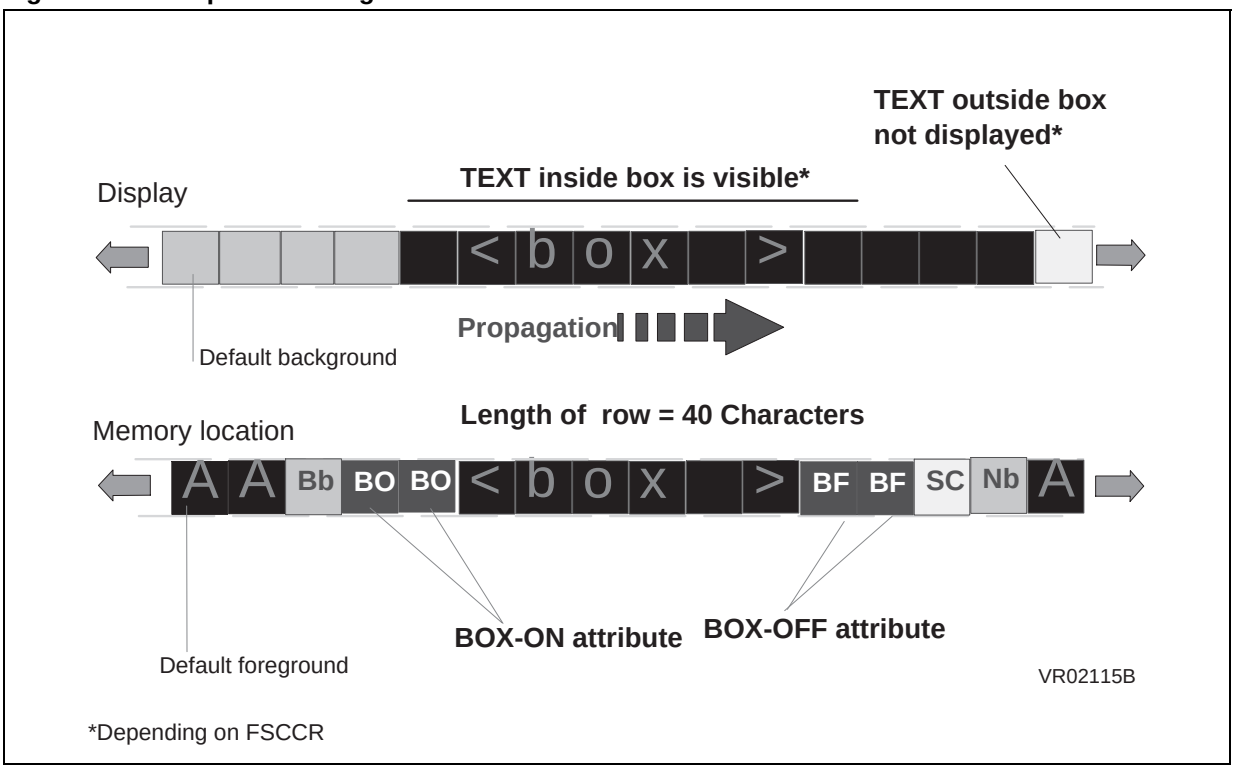
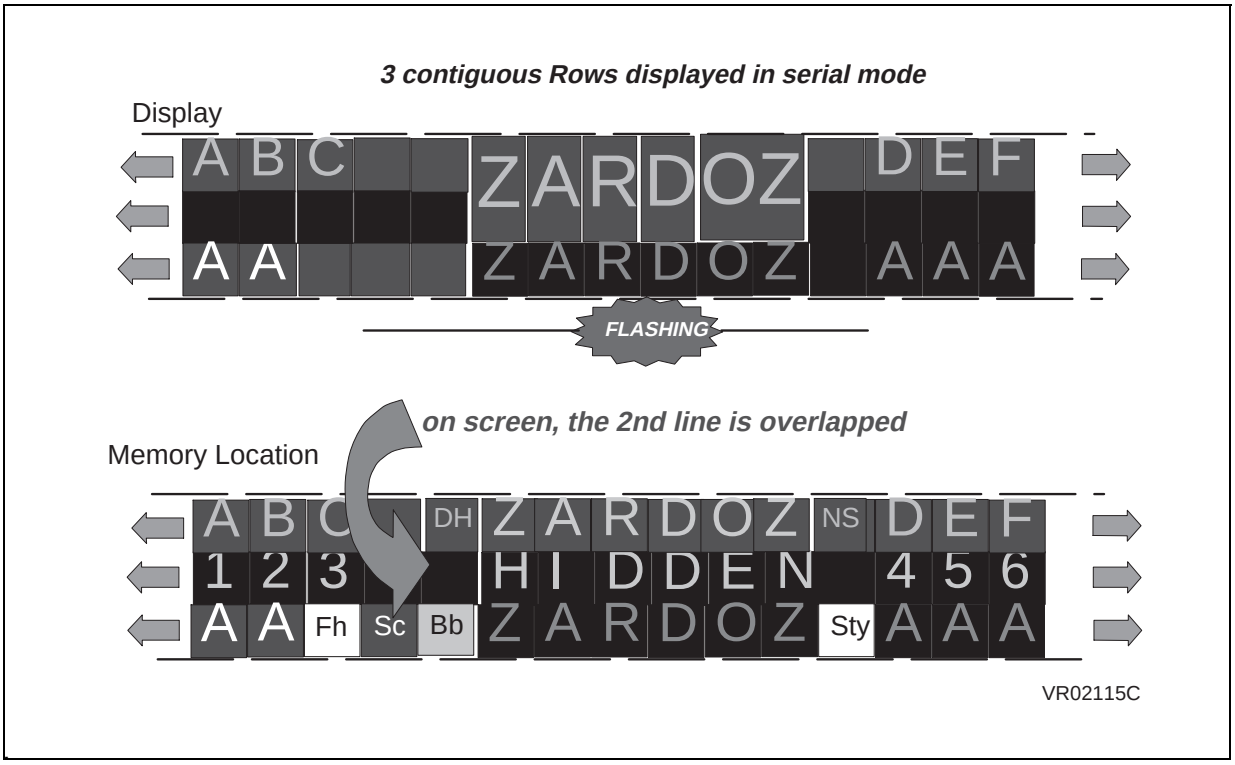


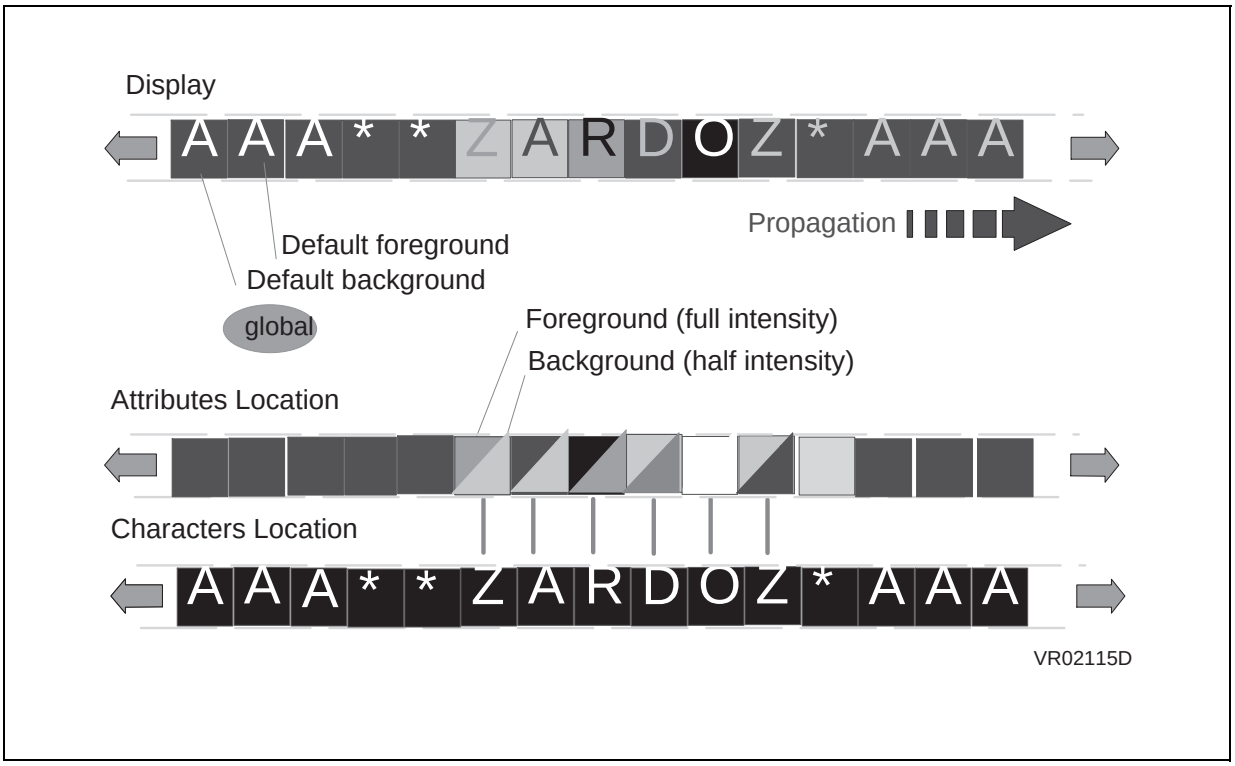
Figure 60. Example of Double Height Attribute in Serial Mode



ON SCREEN DISPLAY (Cont'd)

7.4.4.3 Parallel Attributes

Figure 61. Example of Row in Parallel Mode



Each character is defined on 2 bytes in Parallel Mode (see Figure 61 on page 107.)

Parallel Mode is selected by setting the **SPM** bit in the **DCM1R** register R251 (FBh) Page 32.

It requires 2 bytes per character. Display characters are coded through a second byte processed in parallel with the character code.

It does not handle Teletext and is used mainly for TV menus (e.g. for channel searching or volume control).

The attribute can be one of two types defined by most significant bit (**PS**):

– Color attribute

– Shape attribute

US: Underline / Separate Mosaic graphics (see above).

DH: Double Height: The half character is displayed in the current row depending on the Upper Height control bit. The Double Height action is not propagated in the row.
Note: When a parallel double height attribute is decoded in Row 23, the characters of the first status row are not affected and are still displayed.

UH: Upper Half. This bit is active when the currently displayed row writes the upper half-character in case of double height or double size attribute.

DW: Double Width (see above).

BX: Boxing window.

SR: Smooth Rounding.

FR, FG, FB: Foreground color.

BR, BG, BB: Background color.

HI: Half Intensity (background only).

CSS: Character extended menu code selection.

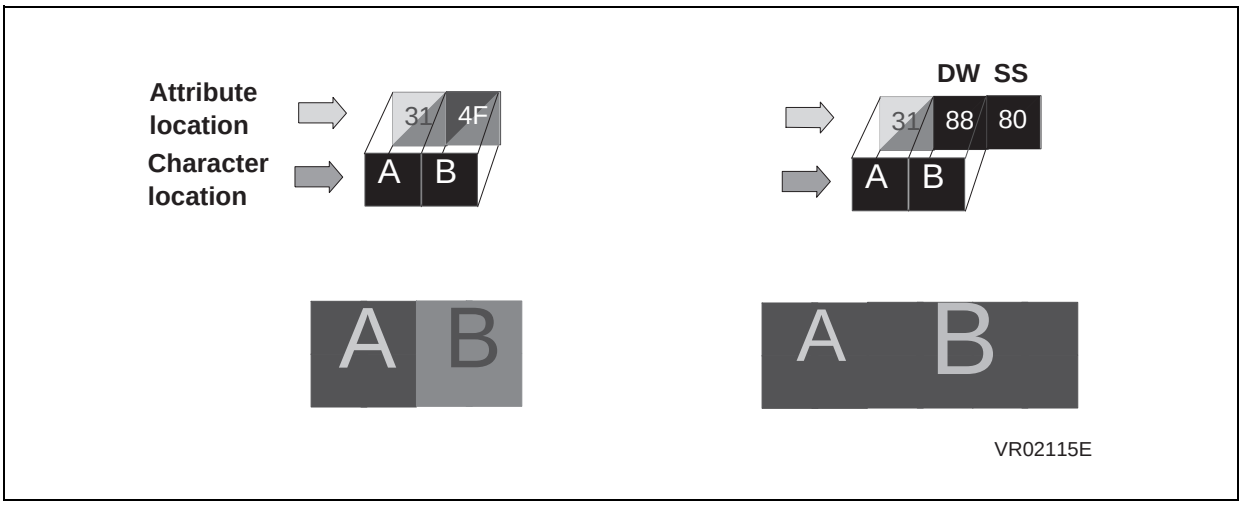
PS: Parallel attribute selection

ON SCREEN DISPLAY (Cont'd)
Table 21. Parallel Color and Shape Attributes.

BIT	NAME	FUNCTION	REMARKS
0	BR	Background Red	
1	BG	Background Green	
2	BB	Background Blue	
3	HI	Half-Intensity	Only for Background.
4	FR	Foreground Red	
5	FG	Foreground Green	
6	FB	Foreground Blue	
7	PS= 0	Parallel Attribute Selection	Color mode of parallel attributes
0	CSS	Character set selection	G2-Menu characters or G1/Extended menu characters selection
1	US	Underline/Seperated mosaic	Dual function depending on character code
2	DH	Double height	The character is 20 pixels high .
3	DW	Double width	The character is 20 pixels wide. Available in Parallel mode or in Line mode. Characters are stretched horizontally, to occupy in addition, the next character space. It is possible to mix it with double height. To display a double width character the attribute must be "double width" on the character and "simple width" on the next which can be a serial attribute. In this case the first character is memorised. If two "double width" attributes are on two adjacent characters, the first half of the second is displayed instead of the second half of the first one.
4	UH	Upper half character (if 1)	Active only if Double Height or Size requested
5	BX	Box mode	Boxing window created (if 1)
6	SR	Smooth rounding	Special rounding effect (See Figure 53 on page 99)
7	PS= 1	Parallel Attribute Selection	Shape mode of parallel attributes

Double Size: (available in Parallel mode or in Line mode) by setting Double Width plus Double Height attributes.

Figure 62. Parallel Color and Shape Attributes



ON SCREEN DISPLAY (Cont'd)

7.4.4.4 Font Selection using Parallel Attributes

Parallel attributes have an immediate effect. They are applied to the associated character. These attributes can also have a "serial" effect, the defined attribute being still defined on the following characters: this is known as attribute propagation.

Shape attributes (US,DH,BX,SR) are propagated when PS is toggled to 0. In the same way, color attributes are propagated when PS is toggled to 1.

CSS has two kinds of behaviour:

– If PS is set once, the CSS attribute is applied on the current character only.

– If PS is set twice, the CSS of the first character with PS=1 is propagated.

Note: The value stored as a preceding CSS value is forced when alpha or mosaic color serial attributes are used. Alpha serial attributes reset the memorized CSS: Mosaic serial attributes set the memorized CSS.

Table 22. Font Selection using Parallel Attributes

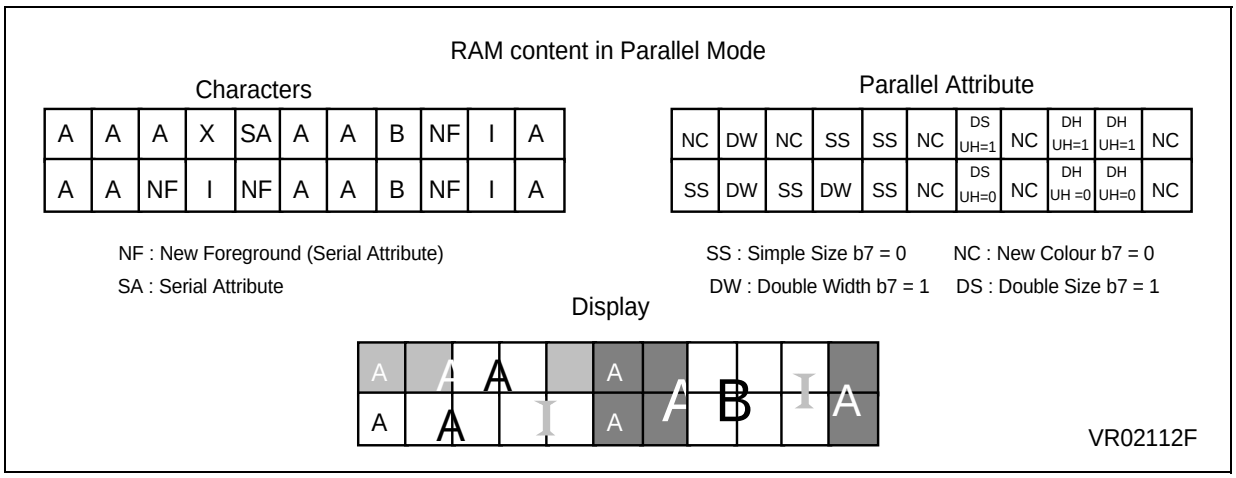
Parallel Attribute	Character Code	Character Definition
PS= 0	00..1F	32 Control Characters (serial attributes function table)
	20..7F	96 Basic Characters chosen from G0 or G1 font
	80..FF	128 extended characters G2-based X/26 and Menu Characters
PS= 1 CSS used for character set selection	00..1F	32 Control Characters (serial attributes function table)
	20..7F	CSS= 0: G0 or G1 selection depending on color serial attribute CSS= 1: G1 selection
	80..FF	CSS= 0: Select G2-based X/26 + Menu CSS= 1: Select extended Menu + 32 reserved characters

In the example in Table 23 Example of Character Set Selection,, a string of six characters is displayed. In the line "Display with" we can see that, starting from Char(n) and ending with Char n+2, the CSS setting made at Char (n-2) is propagated.

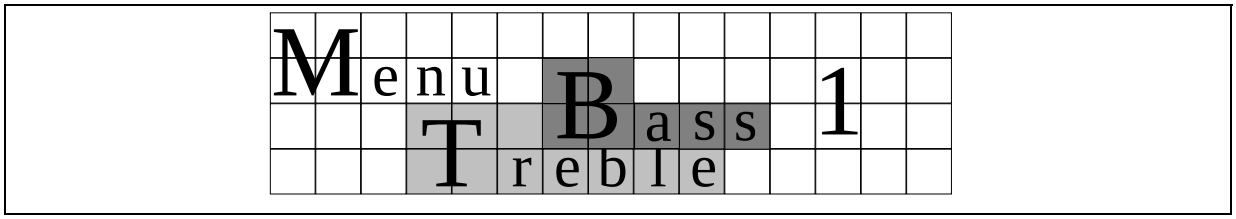
Table 23. Example of Character Set Selection

	Char(n-2)	Char(n-1)	Char(n)	Char(n+1)	Char(n+2)	Char(n+3)
PS=	1	1	0	0	1	1
CSS=	CSSn-2	CSSn-1	none	none	CSSn+2	CSSn+3
Display with	CSSn-2	CSSn-1	CSSn-2	CSSn-2	CSSn+2	CSSn+3
Stored CSS	CSSn-2	CSSn-2	CSSn-2	CSSn-2	CSSn-2	CSSn+2

Figure 63l. Parallel Mode Display Example 1 Showing Character and Attribute Byte Pairs



Parallel Mode Display Example 2



7.4.4.5 Rules When Using Size Attributes

Secondary effects can be generated when the shape format is not respected.

The 3 figures below describe the combination of parallel size attributes to obtain the different character sizes:

· Double Width

· Double Height

· Double Size

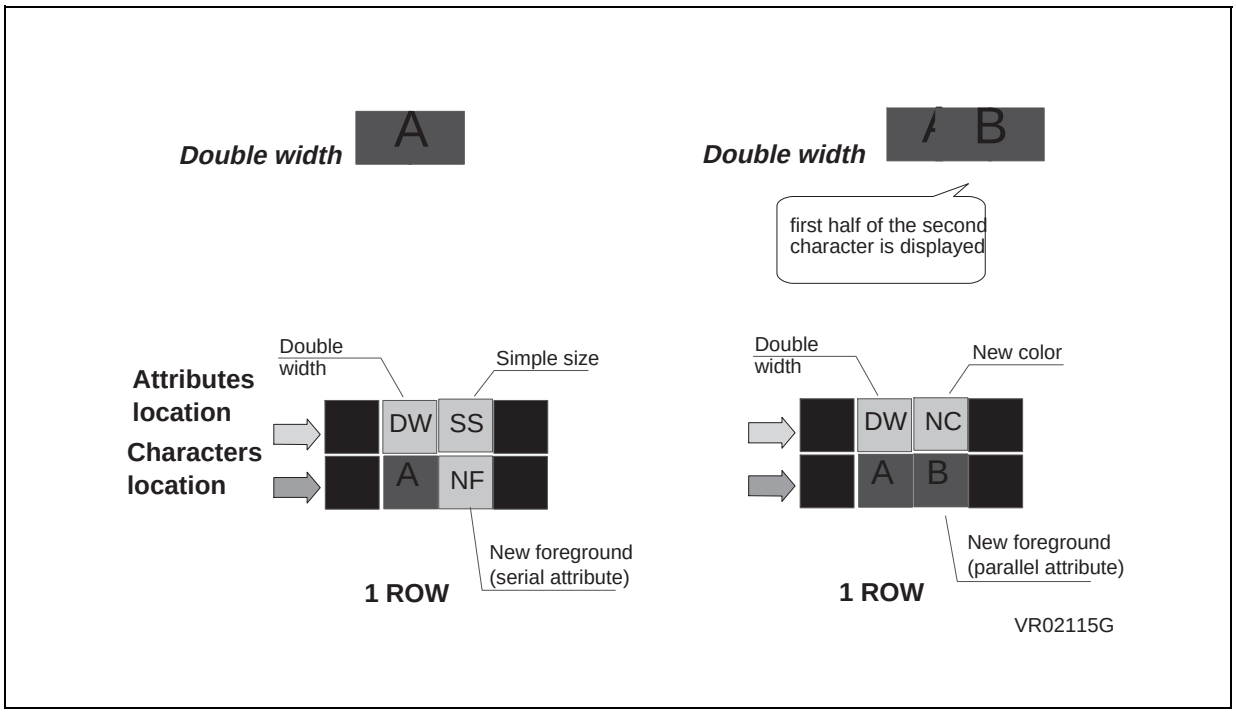
7.4.4.6 Example of using Double Width Attribute

In parallel mode, double width on character can be obtained using the following rule (Figure 64 on page 111):

It is important to set Double Width (bit 3 of the shape attribute) on the current character attribute and Single Size on the following one. The second character location can be either a serial attribute or another character.

On the contrary, if a new color or a Double Width attribute is set in the second attribute location, the second part of the character is overlapped.

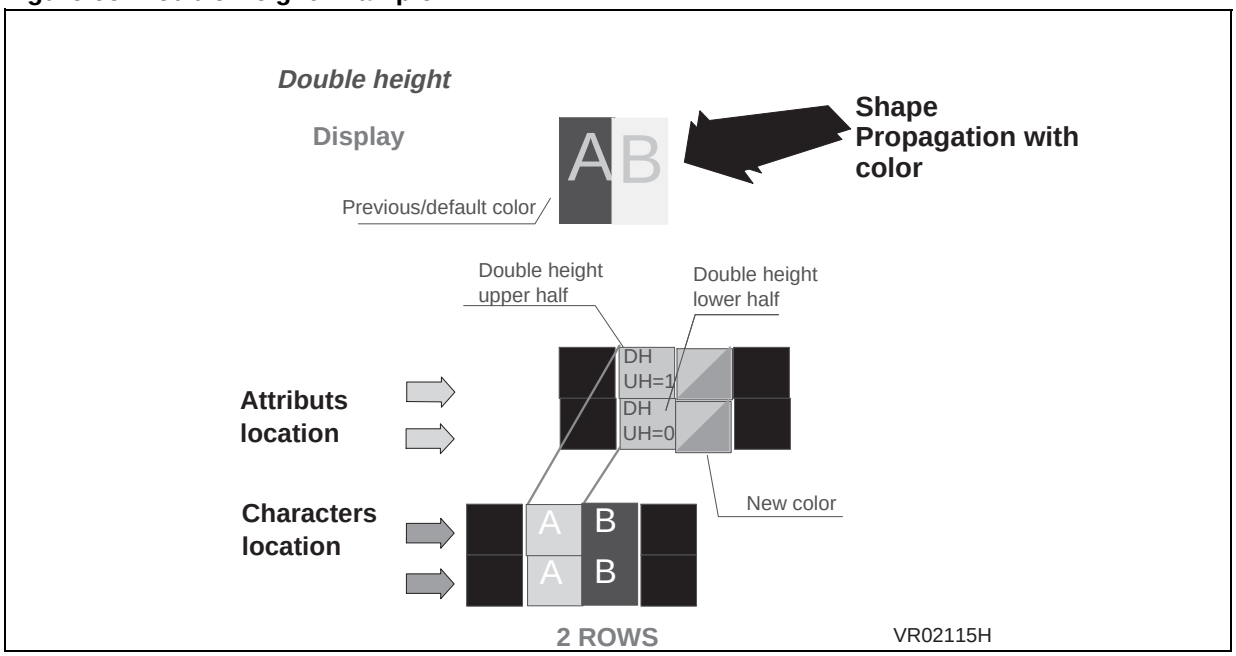
Figure 64. Double Width Examples



7.4.4.7 Example of using Double Height Attribute

In parallel mode, Double Height characters can be obtained as follows. The Double Height attribute concerns two consecutive rows. Repeat the character to magnify in the two rows. Set Bit 2 DH of the shape attribute in the two locations and set or reset bit 4 UH to define if it is the top or bottom half-character.

Figure 65. Double Height Example

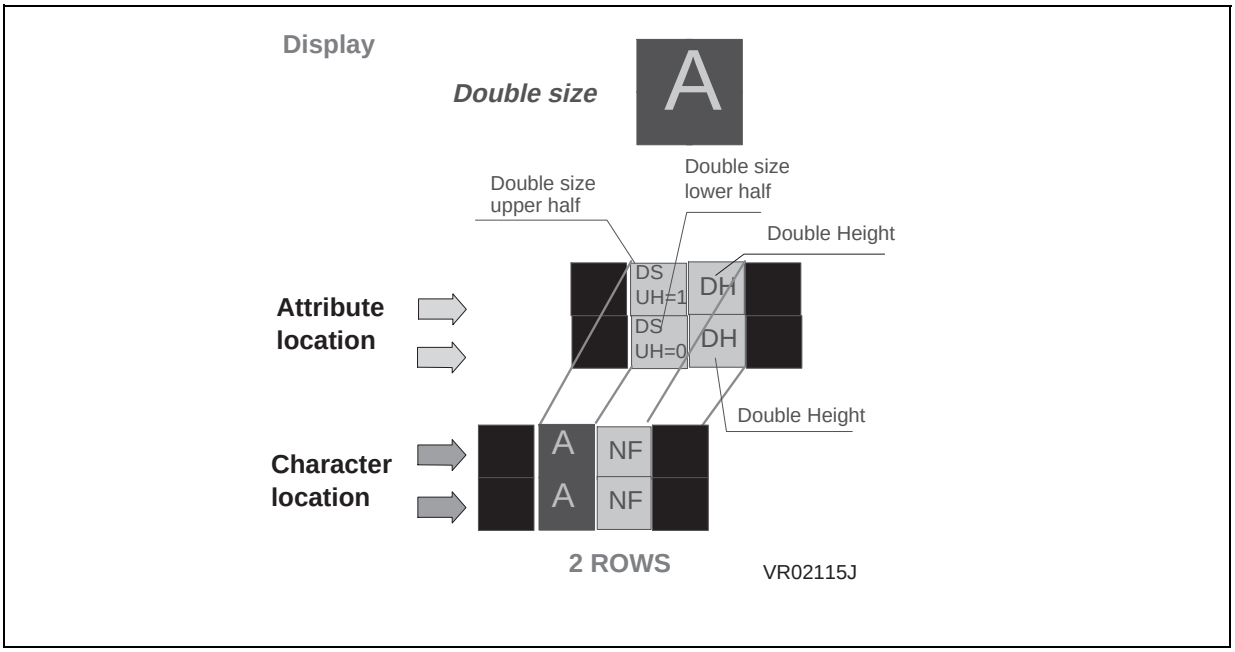


ON SCREEN DISPLAY (Cont'd)

7.4.4.8 Example of using Double Size Attribute

In parallel mode, Double Size characters can be obtained as follows. This attribute concerns two consecutive rows. The character to magnify must be repeated on the two rows. Bits 2 and 3 of the shape attribute must be set on the two locations. In addition bit 4 must be set or reset to define the top or bottom half-character.

Figure 66. Double Size Examples

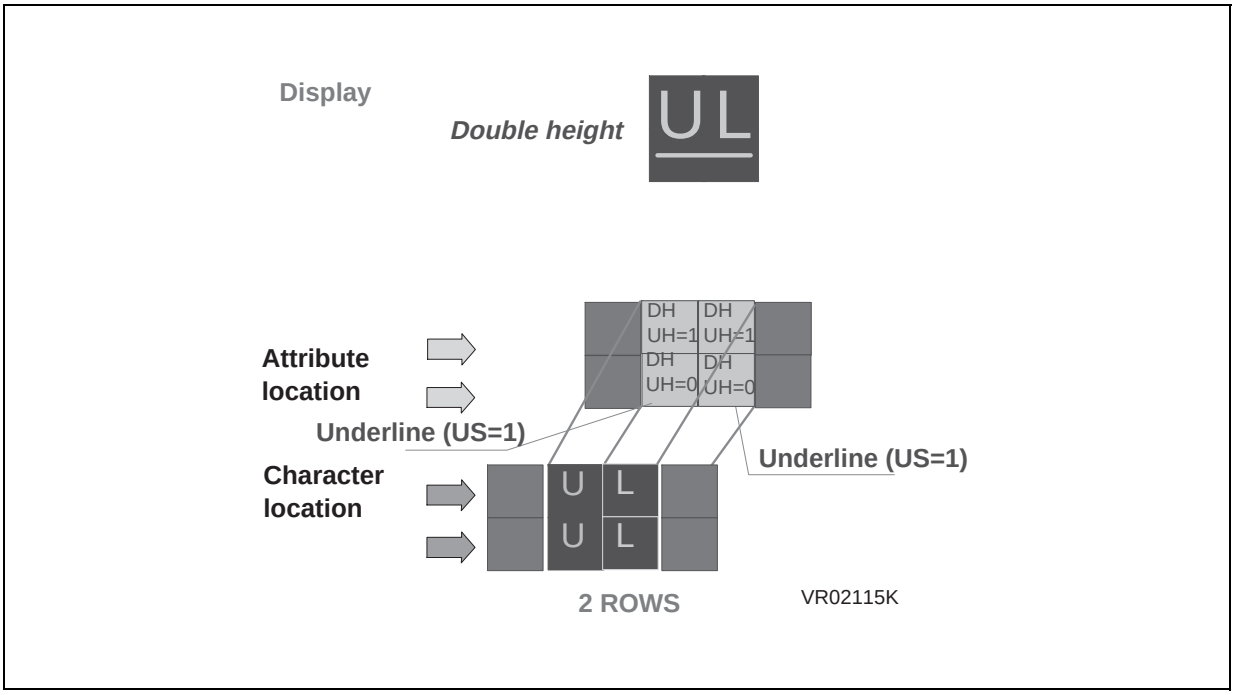


7.4.4.9 Example of using Underline Attribute

In parallel mode, the Underline mode on character can be obtained simply by setting the bit 1 'US' of the shape attribute. To underline double height characters, set the US bit on the attribute associated with the lower part of the character.

The underline attribute is ignored in the upper half-character.

Figure 67. Underline Example



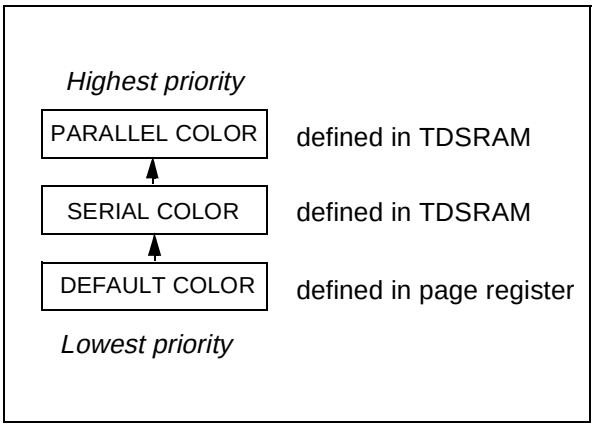
7.4.4.10 Attribute Rules

The default colors for foreground and background are defined through the register **DCR** R240 (F0h) Page 33.

A display defined in parallel mode can accept a serial color attribute, and propagation is available until a new color attribute (serial or parallel) is encountered.

- Rule for Shape Attributes:
  - In parallel mode, shape attributes are not propagated on the following characters of the row except if this character has a colour attribute. The propagation lasts as long as a colour attribute is applied to a character.
  - In parallel mode, the double height (bit 2 of the shape attribute) is active only on its own character. Setting one double height attribute does not cover the following characters of the row (different from double height in serial mode).

Figure 68. Rule for Serial and Parallel Color Combination



ON SCREEN DISPLAY (Cont'd)

7.4.4.11 Cursor Control

- Horizontal position (by character)
- Vertical position (by row)
- Color or Underline Cursor Modes
- Color Cursor with inverted foreground / inverted background
- Flash or Steady mode Color Cursor

Cursor display is controlled using two registers:

– Cursor Horizontal Position R246 (F6h) Page 32

– Cursor Vertical Position R247 (F7h) Page 32.

Notes:

1. Cursor operation in "Underline" mode: any screen location where the foreground color is identical to the background color behaves as a "lost cursor" (i.e. cursor not visible). Assuming a serial mode display, the screen location placed on the lower row after a double height character will lead to a "lost cursor".
2. Ghost fringing: assuming a cursor operation in color inversion mode, assuming a serial mode display, assuming the fringe is activated, the screen location placed on the lower row after a double height character may show a "ghost fringing" effect (the ghost color being an inverted background one).
3. Static or flash cursor Mode: the horizontal cursor value indicates the character position (i.e. first character pointed with a "1" value); in Underline Mode, the horizontal cursor value gives the position minus "1".

7.4.5 Vertical Scrolling Control

- Top-Down or Bottom-Up shift
- Freeze Display function
- Shift speed control
- Double Height Display scrolling

Scrolling is performed in a programmable rolling window if the characters are in normal height.

In Line mode, the scrolling window must be entirely filled by programmed rows (each scrolled location is defined by one of the 11 available rows).

Notes:

1. 80-characters combined with scrolling can only be used in Line mode
2. In Parallel (Level 1+) mode, scrolling is possible without serial attributes DS and DH.

Use these two registers to control scrolling:

– Scrolling Control Low R248 (F8h) Page 32

– Scrolling Control High R249 (F9h) Page 32

7.4.5.1 RGB & FB DAC and TSLU Outputs

The RGB and FB pins of the ST92195C/D are analog outputs controlled by true Digital to Analog Converters. These outputs are specially designed to directly drive the Video Processor.

The effective DAC output level is controlled by a 3 bit digital code issued by the display control logic with respect to the real time value of R, G or B and the Half-Intensity control bit, as follows:

R/G/B DAC code			Display aspect during FB
0	0	0	No Color
0	1	1	Half-Intensity Color
1	1	1	Full-Intensity Color

The analog FB output provides the best phase matching for RGB signals.

An example of the Fast Blanking Signal is shown in Figure 54 on page 102.

The TLSU pin is a digital output (0-5V).

ON SCREEN DISPLAY (Cont'd)

7.4.6 Display Memory Mapping Examples

The display content is stored in TDSRAM, (2 to 8K bytes starting at address 8000h). Use register **TDPR** R252 (FCh) Page 32 to address the memory blocks containing the display data. Two 4-bit address pointers (bits PG and HS) must be given that point to separate blocks containing the display page and the header/status rows.

Alternatively, the PG and HS pointers can be written to the **TDPPR** R246 Page 33 and **TDHSPR** R247 Page 33 registers.

7.4.6.1 Building a Serial Mode Full Page 40-Char Display

Page Location:

The 1 Kbyte block of page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits.

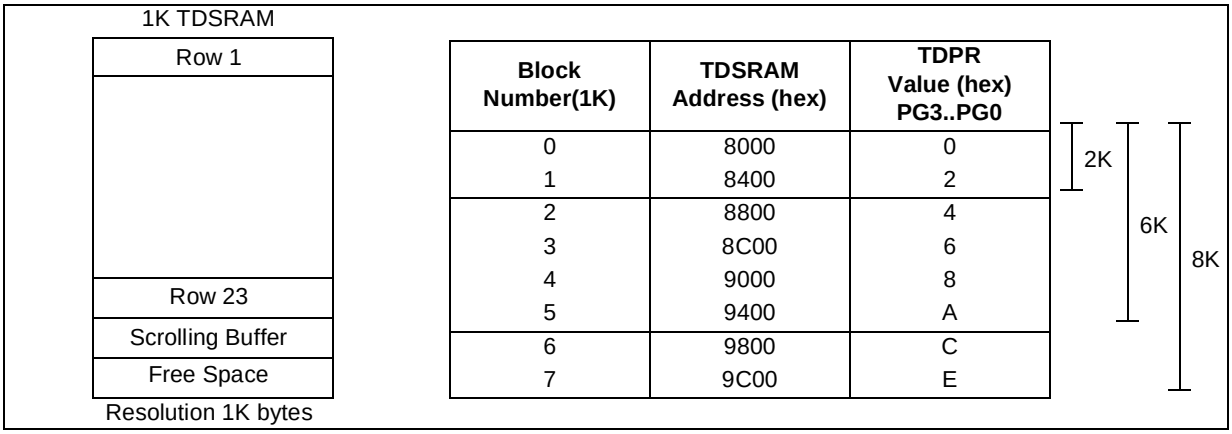
Header & Status Rows Location:

The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits.

Row Scrolling Buffer Location:

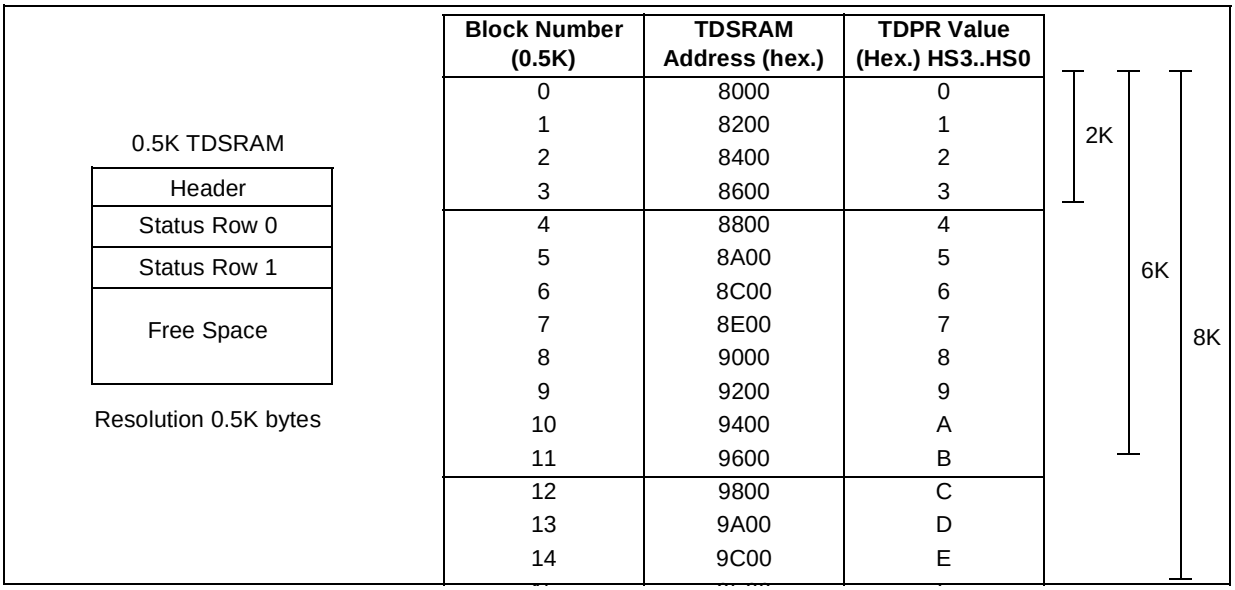
The scrolling buffer corresponds to the 40 bytes following the Row 23 when the scrolling feature is used.

Figure 69. Serial Mode (40 Characters) - Page Mapping



Block Number(1K)	TDSRAM Address (hex)	TDPR Value (hex) PG3..PG0
0	8000	0
1	8400	2
2	8800	4
3	8C00	6
4	9000	8
5	9400	A
6	9800	C
7	9C00	E

Figure 70. Serial Mode (40 Characters) - Header and Status Mapping



Block Number (0.5K)	TDSRAM Address (hex.)	TDPR Value (Hex.) HS3..HS0
0	8000	0
1	8200	1
2	8400	2
3	8600	3
4	8800	4
5	8A00	5
6	8C00	6
7	8E00	7
8	9000	8
9	9200	9
10	9400	A
11	9600	B
12	9800	C
13	9A00	D
14	9C00	E

ON SCREEN DISPLAY (Cont'd)

7.4.6.2 Building a Parallel Mode, 40-Char, Full Page Display

Page Location

The pair of adjacent 1 Kbyte blocks of page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits. The first block contains the characters, the second block contains the attribute bytes.

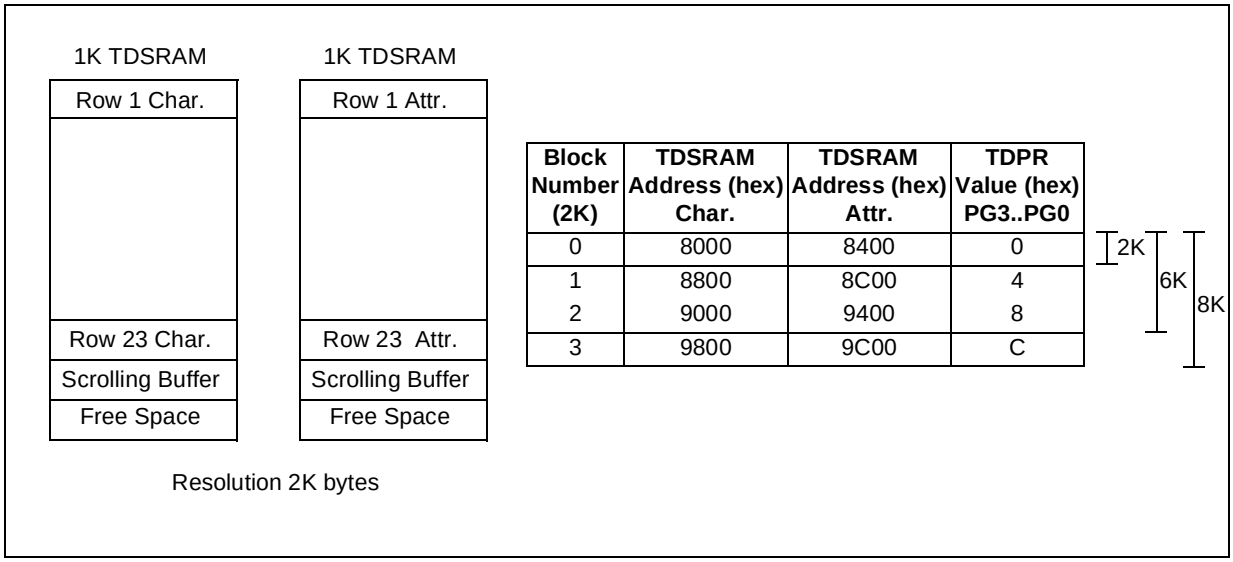
Header & Status Rows Location

The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits. The Header/Status attributes are stored in this block at offset 80h.

Row Scrolling Buffer Location

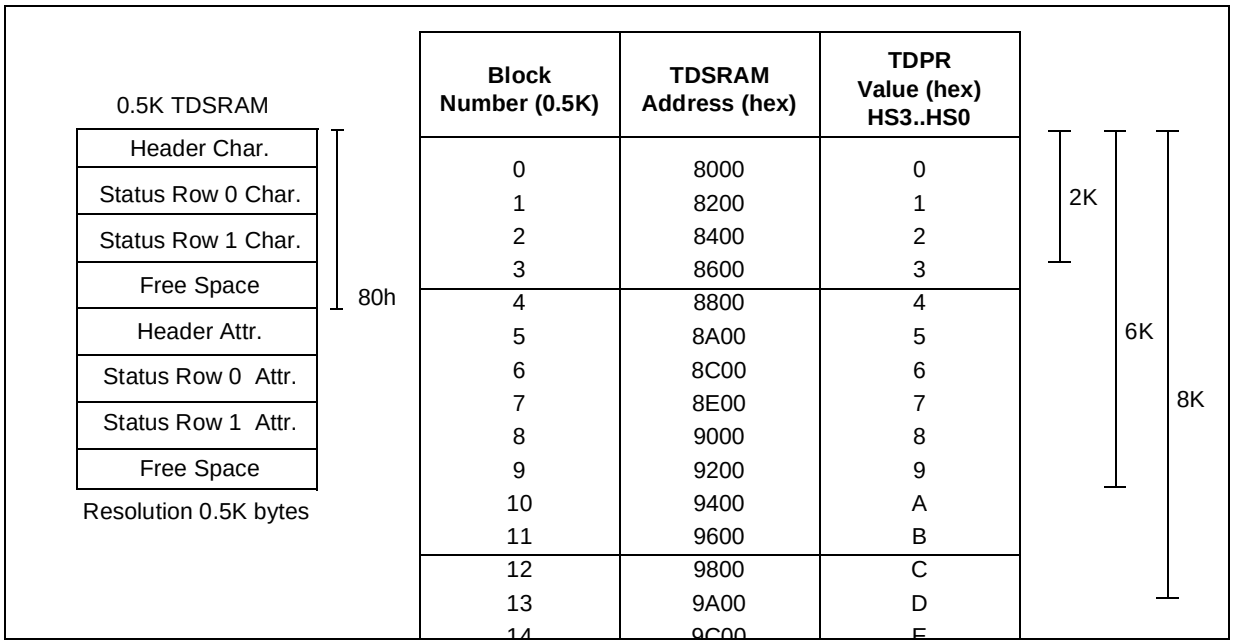
The scrolling buffer corresponds to the 40 bytes following Row 23 when the scrolling feature is used.

Figure 71. Parallel Mode (40 Characters) - Page Mapping



Block Number (2K)	TDSRAM Address (hex) Char.	TDSRAM Address (hex) Attr.	TDPR Value (hex) PG3..PG0
0	8000	8400	0
1	8800	8C00	4
2	9000	9400	8
3	9800	9C00	C

Figure 72. Parallel Mode (40 Characters) - Header and Status Mapping



Block Number (0.5K)	TDSRAM Address (hex)	TDPR Value (hex) HS3..HS0
0	8000	0
1	8200	1
2	8400	2
3	8600	3
4	8800	4
5	8A00	5
6	8C00	6
7	8E00	7
8	9000	8
9	9200	9
10	9400	A
11	9600	B
12	9800	C
13	9A00	D
14	9C00	E

ON SCREEN DISPLAY (Cont'd)

7.4.6.3 Building a Serial Mode, 40-Char, Line Mode Display

Half-Page Location

The 0.5 Kbyte block of half-page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits.

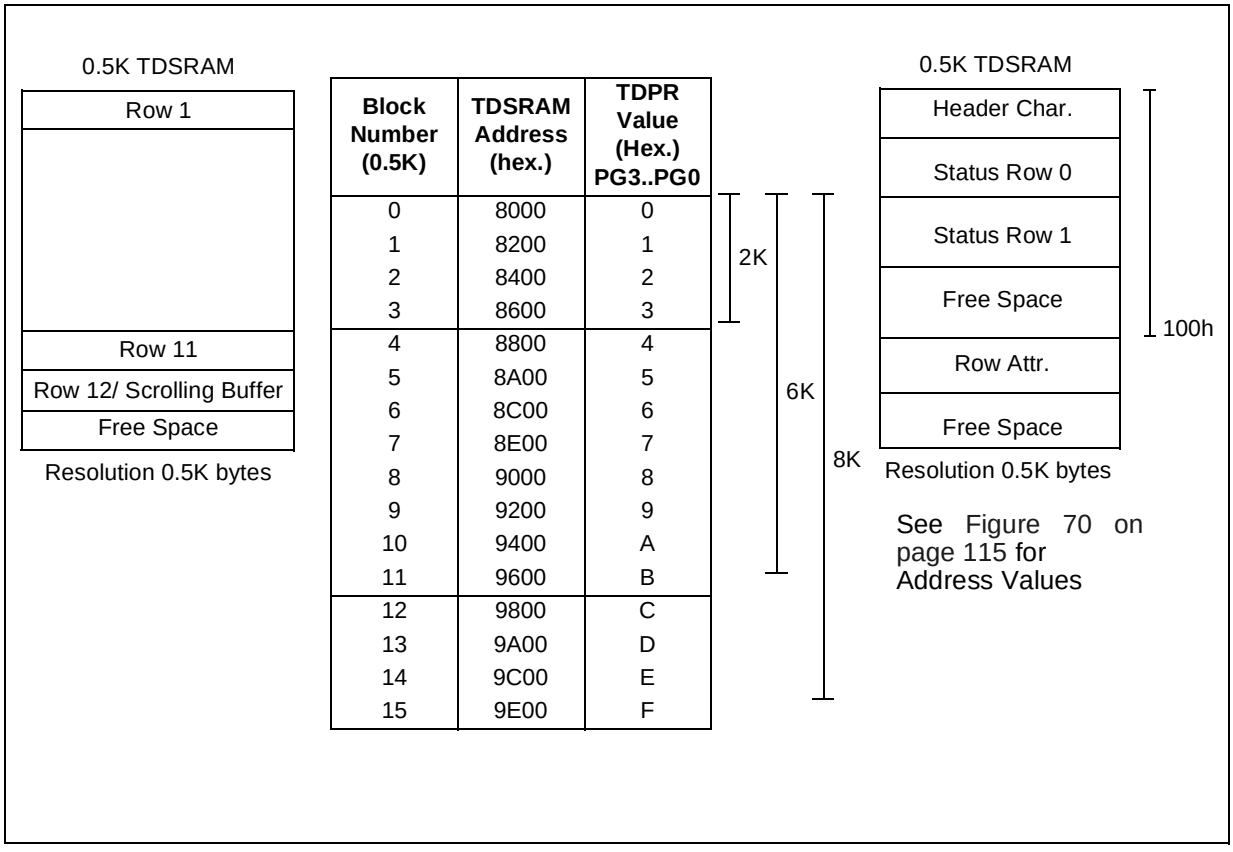
Header & Status Rows Location

The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits. The Row attribute (row count) is stored in this block at offset 100h and contains 12 bytes for line mode (see DCM1R register description).

Row Scrolling Buffer Location

The scrolling buffer corresponds to Row 12 when the scrolling feature is used (in this case 11 rows are scrolled).

Figure 73. Serial (40 Characters) Line Mode Mapping



ON SCREEN DISPLAY (Cont'd)

7.4.6.4 Building a Parallel Mode, 40 Char, Line mode Display

Half-Page Location

The pair of adjacent 0.5 Kbyte blocks of half page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits. One block contains the characters, the other block contains the attribute bytes.

Header & Status Rows Location

The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits.

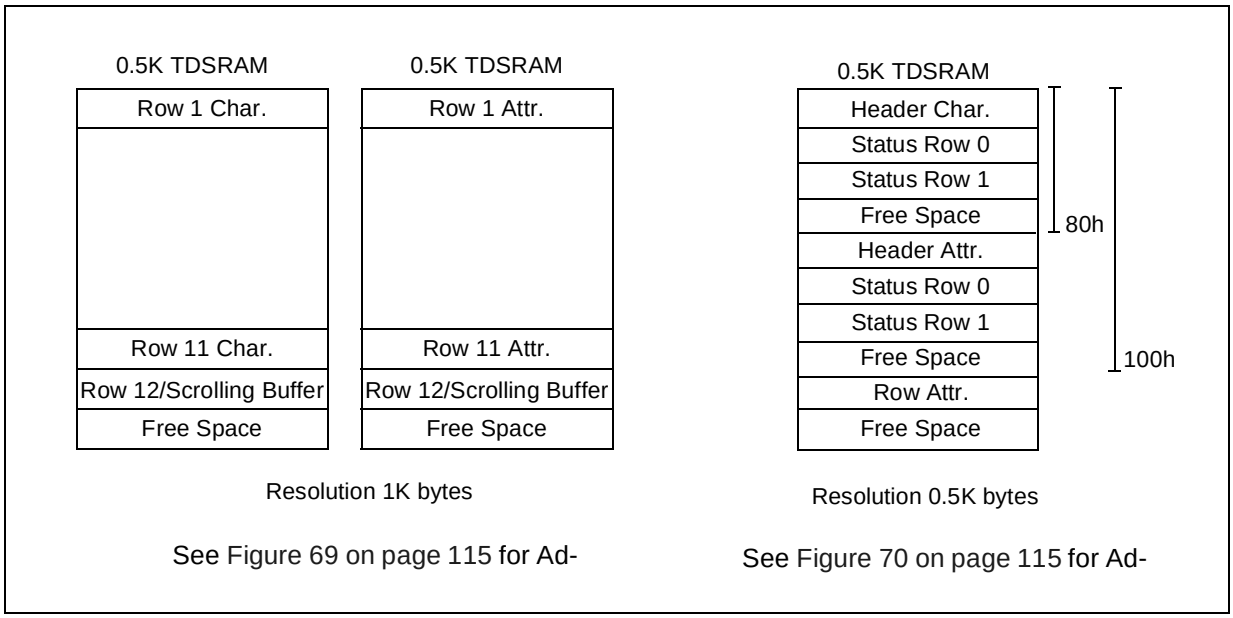
The Header/Status attributes are stored in this block at offset 80h.

The Row attribute (row count) is stored in this block at offset 100h and contains 12 bytes for line mode (see DCM1R register description).

Row Scrolling Buffer Location

The scrolling buffer corresponds to the Row 12 when the scrolling feature is used (in this case 11 rows are scrolled).

Figure 74. Parallel (40 Characters) Line Mode Mapping



ON SCREEN DISPLAY (Cont'd)

7.4.6.5 Building a Serial Mode, 80 Char, Full Page Display

Half-Page Location

The pair of adjacent 1 Kbyte blocks of page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits. The first block contains the left side of the page, the second block contains the right side of the page.

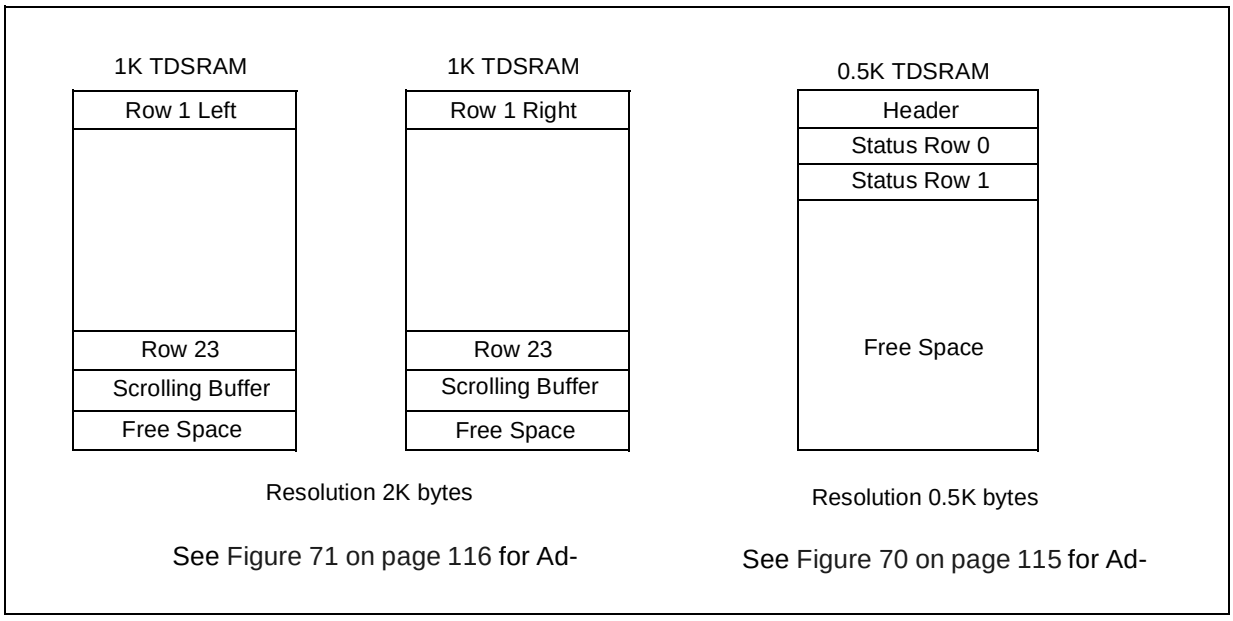
Header & Status Rows Location

The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits.

Row Scrolling Buffer Location

The scrolling buffer corresponds to the 40 bytes following the Row 23 when the scrolling feature is used.

Figure 75. Serial Mode (80 Characters) - Page Mapping



ON SCREEN DISPLAY (Cont'd)

7.4.6.6 Building a Serial Mode, 80 Char, Line Mode Display

Half-Page Location

The pair of 0.5 Kbyte blocks of half page content is stored in the TDSRAM location pointed to by the PG3..PG0 bits. The first block contains the left half rows, the other block contains the right half rows.

Header/Status Rows Location

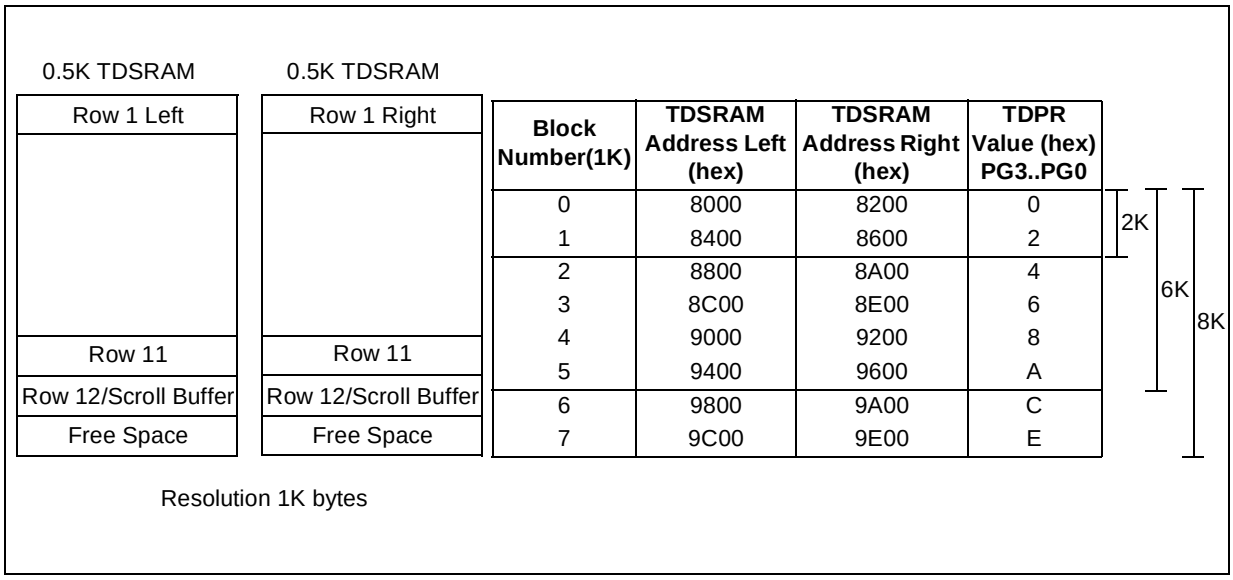
The 0.5 Kbyte block containing the Header, Status Row 0 and Status Row 1 is pointed to by the HS3..HS0 bits.

The Row attribute (row count) is stored in this block at offset 100h and contains 12 bytes for line mode (see DCM1R register description).

Row Scrolling Buffer Location

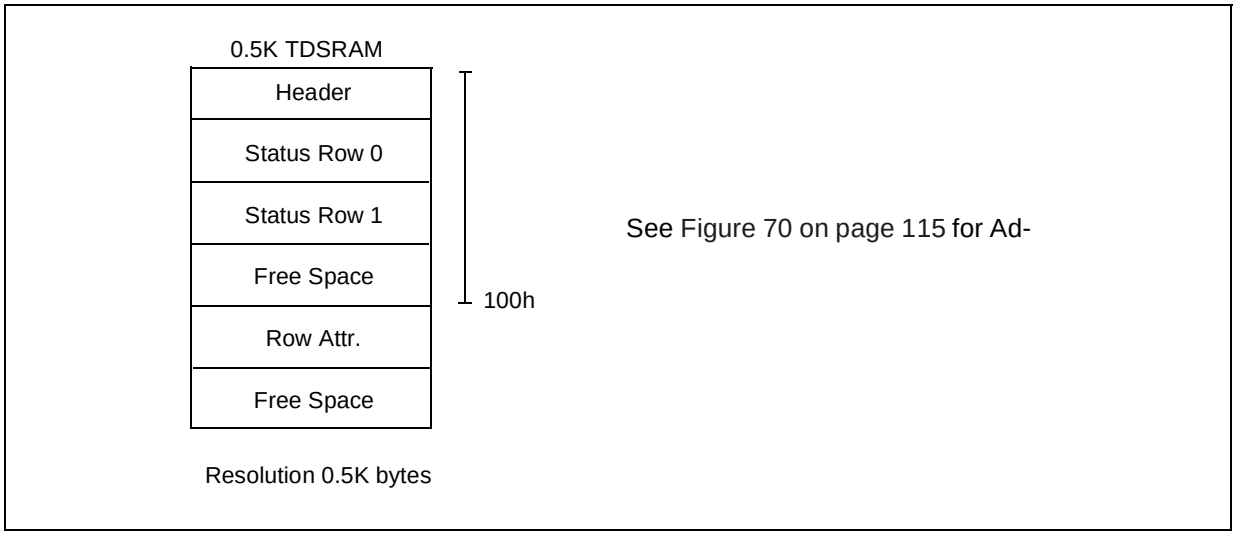
The scrolling buffer corresponds to Row 12 when the scrolling feature is used (in this case 11 rows are scrolled).

Figure 76. Serial (80 Characters) Line Mode Mapping



Block Number(1K)	TDSRAM Address Left (hex)	TDSRAM Address Right (hex)	TDPR Value (hex) PG3..PG0
0	8000	8200	0
1	8400	8600	2
2	8800	8A00	4
3	8C00	8E00	6
4	9000	9200	8
5	9400	9600	A
6	9800	9A00	C
7	9C00	9E00	E

Figure 77. Serial (80 Characters) Line Mode - Header and Status Mapping



See Figure 70 on page 115 for Ad-

ON SCREEN DISPLAY (Cont'd)

7.4.7 Font Mapping

The font consists of two character sets of 512 characters each. Only one set can be accessed at the same time. The FRSW bit of the NCSR register allows to switch from one set to the other.

G0 is the basic character font.

G1 is the mosaic font. It is not stored in ROM but is implemented in hardware. In serial mode it is addressed by a serial attribute (See Table 20. on page 104). In parallel mode it is accessed by bit 0 (CSS) of the parallel shape attribute and bit 1 (US) for separated mosaic (See Table 21. on page 108).

G2 is a font of X/26 based + Menu shared characters.

An Extended Menu character font available in parallel mode. It is accessed via bit 0 (CSS) in the parallel shape attribute (Character Set Selection). The Extended Menu font is not accessible in serial mode.

7.4.8 Font Mapping Modes

There are two font mapping modes selected by the **NCM** bit in the **NCSR** register R245 (F5h) Page 32:

Single G0 mode

A set combining 83 characters from the G0 basic set plus 13 characters selected from 15 National character subsets. The National character subsets are selected by four bits (NC3:0) in the **NCSR** register R245 (F5h) Page 32.

Triple G0 mode

Three 96-character character sets (G0-0, G0-1 and G0-2) for multi alphabet applications. Character set selection is done by four bits (NC1:0 or NC3:2) in the **NCSR** register R245 (F5h) Page 32.

– In Serial Mode (Level 1), only 256 Character Codes are available using an 8-bit code. The character codes plus some serial attributes and some additional programmable options address 566 chars: 256 + 182 NS chars + 128 mosaics in single G0 mode.

– In Parallel Mode (Enhanced Level 1), 512 Character Codes are available using a 9-bit code. The character codes plus some serial and parallel attributes, and some additional programmable options address 662 chars: 256 + 182 NS chars + 128 mosaic + 96 extended chars. in single G0 mode.

Display ROM Font Entry:

The user must define his own fonts for each set of characters:

– 278 characters: - 15 x 13 G0 National Character subsets + 83 G0 Character set

or

– 288 characters: 3 x 96-character character sets

– 128 G2 based X/26 and Menu characters

– 96 Extended Menu characters

Table 24. Triple G0 Mode - Font Mapping

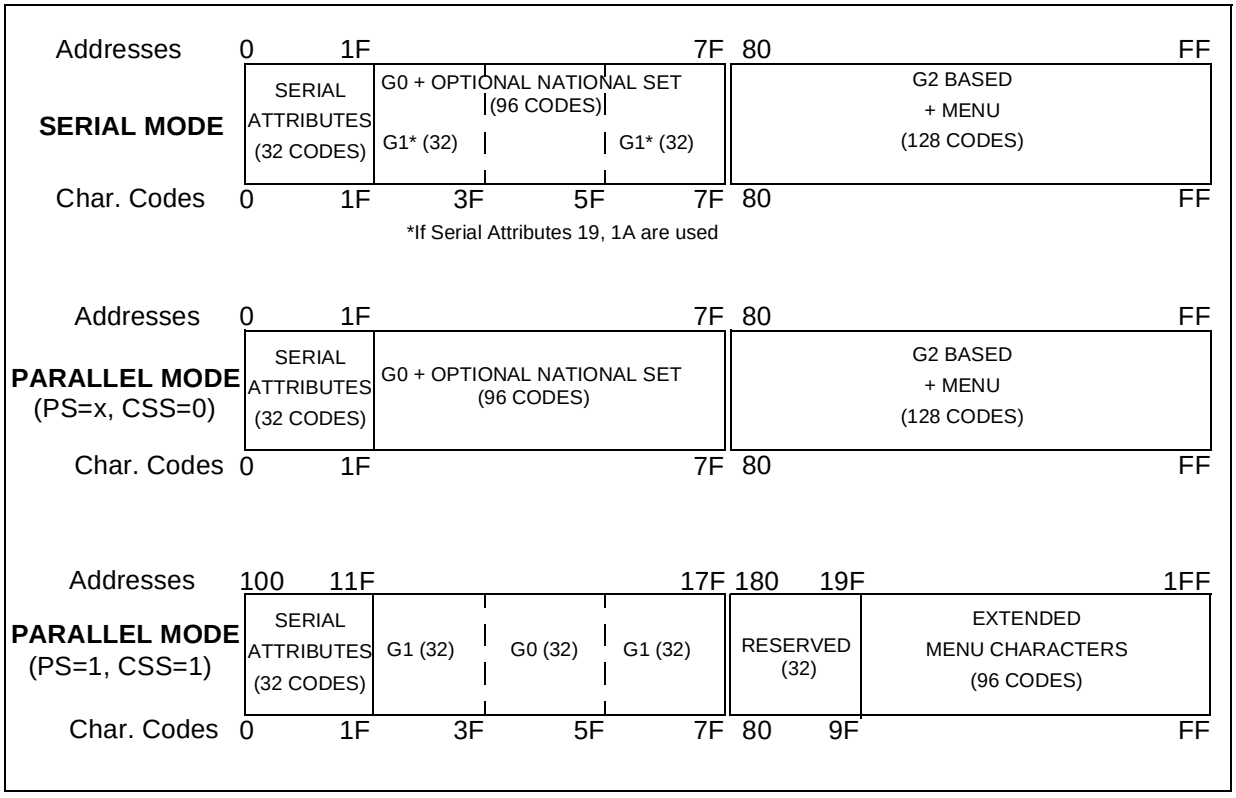
ROM Address	Character Code	CSS	Font Usage
000h to 01Fh	0E0h to 0FFh	1	Extended menu
020h to 07Fh	020h to 07Fh	0 (or serial mode)	G0 set 0
080h to 0FFh	080h to 0FFh		G2 + Menu
100h to 15Fh	020h to 07Fh		G0 set 1
160h to 1BFh	020h to 07Fh		G0 set 2
1C0h to 1FFh	0A0h to 0DFh	1	Extended menu

ON SCREEN DISPLAY (Cont'd)

Table 25. National Character Subset Mapping (Ordered by their G0 address)

1st	2nd	3rd	4th	5th	6th	7th
23h	24h	40h	5Bh	5Ch	5Dh	5Eh
8th	9th	10th	11th	12th	13th	
5Fh	60h	7Bh	7Ch	7Dh	7Eh	

Figure 78. Font Mapping



ON SCREEN DISPLAY (Cont'd)

Table 26. Single G0 Mode - Font Mapping

ROM Address	Character Code	CSS	Font Usage	NC(3:0)
000h to 01Fh	0E0h to 0FFh	1	Extended menu	
020h to 07Fh	020h to 07Fh	0	G0 + National Character Subset 0 (96 chars)	0000b
080h to 0FFh	080h to 0FFh		G2 + Menu (128 chars)	
100h to 10Ch	(see table below)		National Character Subset 1 (13 chars)	0001b
10Dh to 119h	(see table below)		National Character Subset 2 (13 chars)	0010b
11Ah to 126h	(see table below)		National Character Subset 3 (13 chars)	0011b
127h to 133h	(see table below)		National Character Subset 4 (13 chars)	0100b
134h to 140h	(see table below)		National Character Subset 5 (13 chars)	0101b
141h to 14Dh	(see table below)		National Character Subset 6 (13 chars)	0110b
14Eh to 15Ah	(see table below)		National Character Subset 7 (13 chars)	0111b
15Bh to 167h	(see table below)		National Character Subset 8 (13 chars)	1000b
168h to 174h	(see table below)		National Character Subset 9 (13 chars)	1001b
175h to 181h	(see table below)		National Character Subset 10 (13 chars)	1010b
182h to 18Eh	(see table below)		National Character Subset 11 (13 chars)	1011b
18Fh to 19Bh	(see table below)		National Character Subset 12 (13 chars)	1100b
19Ch to 1A8h	(see table below)		National Character Subset 13 (13 chars)(Free for user)	1101b
1A9h to 1B5h	(see table below)		National Character Subset 14 (13 chars) (Menu chars.)	1110b
1C0h to 1FFh	0A0h to 0DFh	1	Extended menu	

Table 27. National Character Subsets

Subset Name	Subset No. (Decimal)	Character Code (Hex)												
		23	24	40	5B	5C	5D	5E	5F	60	7B	7C	7D	7E
Czech/Slovak	3	#	ů	č	ť	ž	ý	í	ř	é	á	ě	ú	š
English	0	£	$	@	←	½	→	↑	#	—	¼	‖	¾	÷
Estonian	9	#	õ	Š	Ä	Ö	Ž	Ü	Õ	š	ä	ö	ž	ü
French	1	é	ï	à	ë	ê	ù	î	#	è	â	ô	û	ç
German	4	#	$	§	Ä	Ö	Ü	^	_	°	ä	ö	ü	ß
Italian	6	£	$	é	°	ç	→	↑	#	ù	à	ò	è	ì
Lettish/ Lithuanian	10	#	$	Š	ė	ę	Ž	č	ū	š	ą	ų	ž	į
Polish	8	#	ń	ą	Ż	Ś	Ł	ć	ó	ę	ż	ś	ł	ź
Portugese/ Spanish	5	ç	$	¡	á	é	í	ó	ú	¿	ü	ñ	è	à
Rumanian	7	#	¤	Ţ	Â	Ş	Ă	Î	ı	ţ	â	ş	ă	î
Serbian/ Croatian/ Slovenian	12	#	Ë	Č	Ć	Ž	Đ	Š	ë	č	ć	ž	đ	š
Swedish/ Finnish	2	#	¤	É	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
Turkish	11	₺	ğ	İ	Ş	Ö	Ç	Ü	Ğ	ı	ş	ö	ç	ü

ON SCREEN DISPLAY (Cont'd)

Figure 79. Pan-European Font (East/West) Character Codes (Hex.)

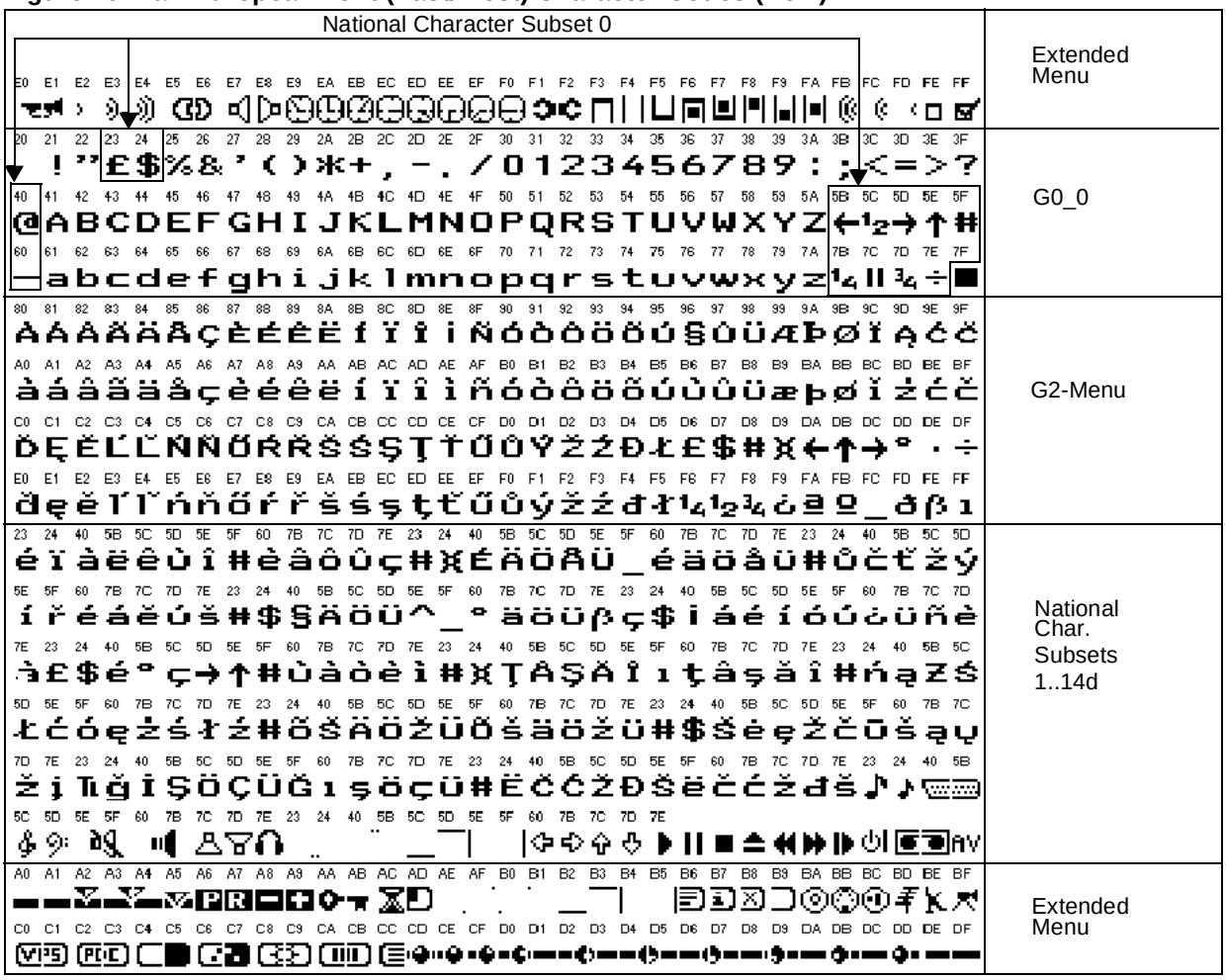


Figure 80. OSD Picture in Parallel Mode



ON SCREEN DISPLAY(Cont'd)

7.4.9 Register Description

HORIZONTAL BLANK REGISTER (HBLANKR)
R240 - Read/Write
Register Page: 32
Reset Value: 0000 0011 (03h)

7							0
HB7	HB6	HB5	HB4	HB3	HB2	HB1	HB0

It controls the length of the Horizontal Blank which follows the horizontal sync pulse.

Bits 7:0 = **HB[7:0]**: The horizontal blank period is calculated with a pixel down counter loaded on each Hsync by HB[7:0]. During this period, FB = 0 and (R, G, B) = black.

Thblank = [(HB7*128 + HB6*64 + HB5*32 + HB4*16 + HB3*8 + HB2*4 + HB1*2 + HB0) * Tpix]

HORIZONTAL POSITION REGISTER (HPOSR)
R241 - Read/Write
Register Page: 32
Reset Value: 0000 0011 (03h)

7							0
HP7	HP6	HP5	HP4	HP3	HP2	HP1	HP0

Bits 7:0 = **HP[7:0]**: The horizontal start position is calculated with a pixel down-counter loaded on each Hsync by HP[7:0]. The first character display starts when the counter turns to zero.

Hori delay= [(HP7*128 + HP6*64 + HP5*32 + HP4*16 + HP3*8 + HP2*4 + HP1*2 + HP0) * Tpix] + Thblank

VERTICAL POSITION REGISTER (VPOSR)
R242 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
0	0	VP5	VP4	VP3	VP2	VP1	VP0

Bits 7:6 = Reserved, keep in reset state.

Bits 5:0 = **VP[5:0]:** The vertical start position is calculated with a line downcounter decremented on each Hsync by VP[5:0]. The Display of the first row begins when the counter turns to zero.

Vert delay = (VP5*32 + VP4*16 + VP3*8 + VP2*4 + VP1*2 + VP0) * Tline (Tline= 64 µs)

ON SCREEN DISPLAY (Cont'd)

FULL SCREEN COLOR CONTROL REGISTER (FSCCR)
R243 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
BE	TIO	MM	HTC	FSC3	FSC2	FSC1	FSC0

Bit 7 = **BE**: Box Enable, see Table 28 Box Mode/ Translucency Configurations.

Bit 6 = **TIO**: Text out/not in, see Table 28 Box Mode/Translucency Configurations.

Bit 5 = **MM**: Mixed Mode, see Table 28 Box Mode/ Translucency Configurations.

Note: When Flash and Box attributes are decoded at the same time on the characters of a header (when BE=1, MM=1, TIO=1) the full screen over the characters is displayed as transparant.

Bit 4 = **HTC**: *Half-Tone/Translucency Control Bit*
This bit allows the selection of TSLU or HT as alternate function output.
0: TSLU is selected as I/O pin alternate function
1: HT is selected as I/O pin alternate function

Bits 3:0 = **FSC[3:0]**: *Full Screen Color control bits*:
FSC[3:0]= (Half-intensity, R, G, B)

Table of Color Values (hex)

0 Black	8 Black
1 Blue	9 Dark blue
2 Green	A Dark green
3 Cyan	B Dark cyan
4 Red	C Dark red
5 Magenta	D Dark magenta
6 Yellow	E Dark yellow
7 White	F Grey

Table 28. Box Mode/Translucency Configurations

BE	TIO	MM	If Translucency is not used	If Translucency is used
0	x	0	Solid Background for all the display	Translucent Background for all the display
0	x	1	Transparent Background for all the display	Transparent Background for all the display
1	0	0	Text inside box solid, Text outside box blanked	Text inside box translucent, Text outside box blanked
1	0	1	Text inside box with solid background Text outside box with transparent background	Text inside box with translucent background Text outside box with transparent background
1	1	0	Text inside box not displayed, transparent background. Text outside box with solid background	Text inside box not displayed, transparent background. Text outside box with translucent background
1	1	1	Text inside box with transparent background. Text outside box with solid background	Text inside box with transparent background. Text outside box with translucent background

ON SCREEN DISPLAY (Cont'd)

HEADER & STATUS CONTROL REGISTER (HSCR)
R244 - Read/Write
Register Page: 32
Reset Value: 0010 1010 (2Ah)

7							0
0	0	ES1	NS1	ES0	NS0	EH	NH

Bits 7:6 = Reserved.

Bits 5,3 = **ES[1:0]**: *Enable Status Rows* [1:0] display control bits. If the bit is reset, the corresponding Status Row is filled with the full screen color; if the bit is set, the corresponding Status Row is displayed (Status Row 1 is assumed to be the bottom one).

Bits 4,2 = **NS[1:0]**: *Serial/Parallel Mode Status Rows display control bits*. If the corresponding bit is reset, the Status Row uses only serial attributes. If the corresponding bit is set, the Status Row uses parallel attributes (except size attributes).

Bit 1 = **EH**: *Enable Header display control bit*. If set, the Header row is displayed; if reset, the Header row is filled with the full screen color.

Bit 0 = **NH**: *Serial/Parallel Mode Header display control bit.* If the bit is reset, the Header uses only serial attributes. If the bit is set, the Header uses of parallel attributes.

ON SCREEN DISPLAY (Cont'd)

NATIONAL CHARACTER SET REGISTER (NCSR)
R245 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
TSLE	FRSW	SWE	NCM	NC3	NC2	NC1	NC0

The register bit values are sampled and then activated only at each field start (on Vsync pulse).

Bit 7 = **TSLE:** *Translucency/Half-Tone Output Enable bit.*
0: Translucency/Half-Tone signal disabled
1: Translucency/Half-Tone is enabled. Translucency or Half-Tone realtime control signal is routed in the TSLU/HT pin (depending on the HTC bit in the FSCCR register).

Note: Translucent display depends also on the BE, TIO and MM bits, see Table 28 Box Mode/ Translucency Configurations.

Bit 6 = **FRSW:** *Font ROM Switch bit.*

This bit allows to switch from the first Font ROM set to the second one.
0: When the FRSW bit is reset, the first set of 512 ROM characters (FROM 0) is selected.
1: When the FRSW bit is set, the second set of 512 ROM characters (FROM1) is selected.

Bit 5 = **SWE**: *G0 Switch Enable Control Bit.*
In case of a multiple G0 alphabet configuration (NCM=1), this bit allows to switch from a first to a second predefined G0 alphabet, using a single serial attribute (1Bh). In case of a single G0 alphabet configuration (NCM=0), the SWE bit will have no effect.

If SWE is reset, the used G0 alphabet is pointed through NC[1:0].

If SWE is set, the used G0 alphabet is pointed through NC[3:2] and NC[1:0] toggled by 1Bh serial attribute.

Bit 4 = **NCM**: *National Character Mode control bit.* This bit reconfigures a part of the font set as defining:

– either a single G0 alphabet with up to 15 national sub-sets,

– or 3 different G0 alphabets.

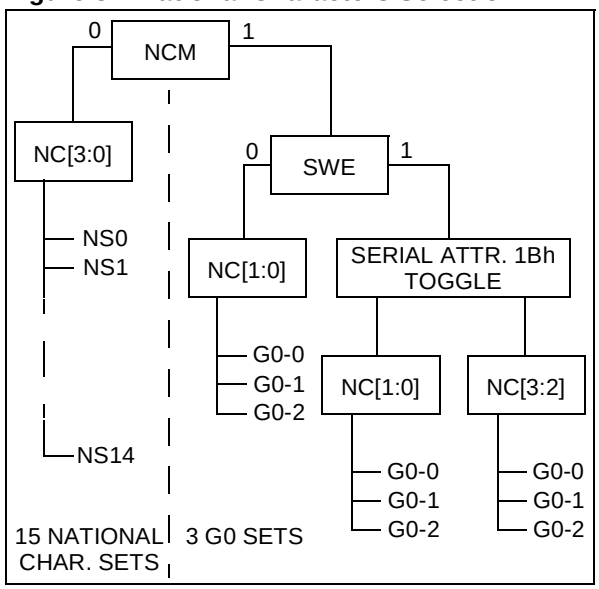
If NCM is reset, a single G0 alphabet configuration is activated and the 15 national sub-sets are selected through the NC[3:0] bits.

If NCM is set, a triple G0 alphabet configuration is activated, the selection of the G0 set used for the display is done through either NC[3:2] or NC[1:0] bits, depending upon the SWE control bit and the serial attribute 1Bh values.

Bits 3:0 = **NC[3:0]**: *National Character Set Selection.*
If the NCM bit is reset, these bits define which national sub-set has to be used to complete the basic currently used G0 alphabet set.

If the NCM is set, these bits define which G0 is used.

Figure 81. National Characters Selection



ON SCREEN DISPLAY (Cont'd)

CURSOR HORIZONTAL POSITION REGISTER (CHPOSR)
R246 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
0	CHP6	CHP5	CHP4	CHP3	CHP2	CHP1	CHP0

Bit 7 = Reserved.

Bits 6:0 = **CHP[6:0]**: *Cursor Horizontal Position*.
The cursor is positioned by character.

CHP= 0 points to the first character
CHP= 39d points to the end of the row (single page display)
CHP= 79d points to the last character in the row (double page display)

CURSOR VERTICAL POSITION REGISTER (CVPOSR)
R247 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
FON	CM1	CM0	CVP4	CVP3	CVP2	CVP1	CVP0

Bit 7 = **FON**: *"Flash On" flag bit.*
The FON bit remains at "0" during 32 consecutive TV fields followed by a "1" state during the 16 next TV fields. This flag provides a 1Hz time reference for an easy software control of all flashing effects (assuming a 50 Hz TV signal, the FON total period will be 0.96 seconds).

This bit is READ ONLY. Trying to write any value will have no effect.

Bits 6:5 = **CM[1:0]:** *Cursor Mode control bits*.

CM1	CM0	Cursor Mode
0	0	Cursor Disable
0	1	Static Cursor (inverted foreground & inverted background colours)
1	0	Flash Cursor (flash from current to inverted colours & vice versa)
1	1	Cursor done with Underline (use of current foreground color)

Bits 4:0 = **CVP[4:0]**: *Cursor Vertical Position*.
The cursor is positioned by row. The cursor is always single size.

CVP= 0 locates the cursor on the Header row

CVP= 25d locates the cursor on the last Status row.

ON SCREEN DISPLAY (Cont'd)

SCROLLING CONTROL LOW REGISTER (SCLR)
R248 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
SCE	FSC	SS	FRS4	FRS3	FRS2	FRS1	FRS0

Bit 7 = **SCE**: *Scrolling Enable*

Before enabling scrolling, the scrolling area must be defined by the FRS[4:0] and LRS[4:0] bits. The scrolling direction is defined by the UP/D bit.
0: Disable scrolling
1: Enable scrolling

Bit 6 = **FSC**: *Freeze scrolling*
Note: The 2 control bits SCE and FSC must be set to "1" before enabling the Global double height (see the DH bit in the SCHR register).

Bit 5 = **SS**: *Scrolling Speed Control bit.*
0: The display is shifted by 2 TV lines at each TV frame (i.e. after 2 Vertical sync pulses).
1: The display is shifted by 4 TV lines at each TV frame.

Bits 4:0 = **FRS[4:0]**: These bits define the uppermost Row value to be scrolled (rows are numbered from 1 to 23). In case of global double height mode programming, FRS[4:0] must mandatorily be equal to 00000.

Table 29. Scrolling Control Bits

DH	SCE	FSC	UP/D	FRS[4:0]	LRS[4:0]	Meaning
0	0	x	x	x	x	No Global Double Height, No Scrolling
0	1	x	1	x	x	No Global Double Height, Scroll up
			0	x	x	No Global Double Height, Scroll down
1	0	x	1	0	x	Global Double Height, No Scrolling, Display top half
			0	0	x	Global Double Height, No Scrolling, Display bottom half

ON SCREEN DISPLAY (Cont'd)

SCROLLING CONTROL HIGH REGISTER (SCHR)
R249 - Read/Write
Register Page: 32
Reset Value: 0000 0000b (00h)

7							0
DH	EER	UP/D	LRS4	LRS3	LRS2	LRS1	LRS0

Bit 7 = **DH**: *Global Double Height control bit*.
This bit must only be used in Page Mode. When DH is set, the display is turned in double height including the header, excluding the vertical offset before the display area. The status rows are not affected by the DH bit and they remain in normal height. Depending on the value of the UP/D control bit, when DH is set, the first or second half of the page is displayed in double height. This bit assumes a zooming function.

Notes:

– In global double height, when the top half page is displayed, if row 11 has a double height attribute, the first status row is corrupted. To avoid this effect, save row 11, remove the serial double height attribute from this row and display the upper part of the page. Then, before displaying the lower part of the page, restore the serial DH attribute in row 11.

– When the bottom half page is displayed, if row 23 has a double height attribute, the first status row is not displayed. To avoid this effect remove the serial double height attribute from row 23.

Bit 6 = **EER**: *End of Extra Row flag bit.*
This bit is forced to "1" by hardware when the last line of the extra row is displayed in case of scrolling in normal height. This bit is Read only.

In Global double height, the EER bit is set to "1" each time the last line of a new displayed row appears.

Bit 5 = **UP/D**: *Scrolling Up/Down*
This bit has two functions: to control the scrolling direction and to select the half part of the page in Global Double Height display.
Scrolling direction:
0: Top-Down shift
1: Bottom-up shift

Half-page selection:

When DH is set, if UP/D is set, the upper half of the page is displayed (i.e. Header and the page rows 1 to 11).

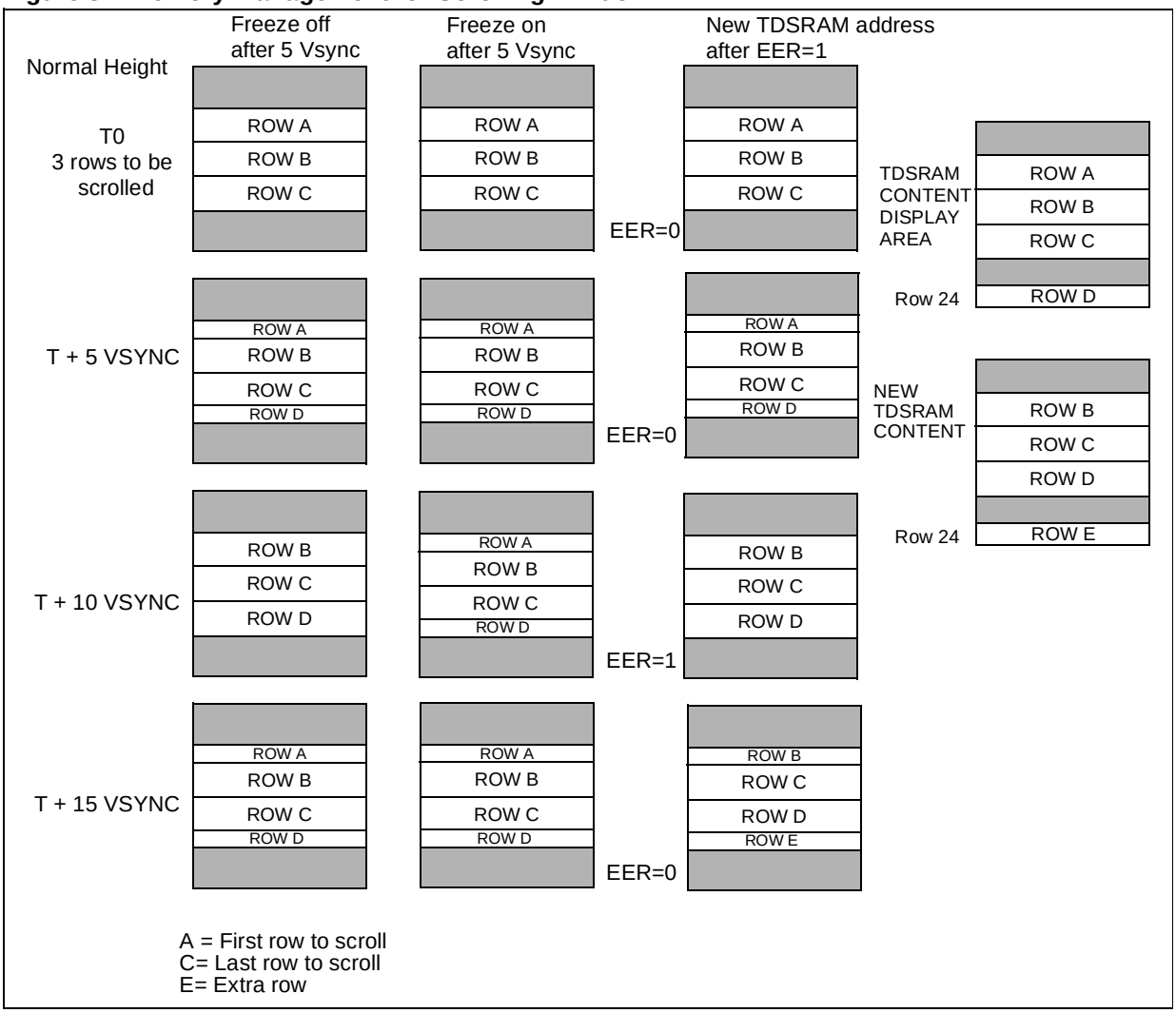
When DH is set, if UP/D is reset, the lower half of the page is displayed (i.e. rows 12 to 23 and the Status rows).

The UP/D control bit must be defined before setting the Global height (DH bit); changing UP/D after DH is set, will not change the already selected half page.

Bits 4:0 = **LRS[4:0]**: *Last row to be scrolled* (1 to 23). In case of scrolling in global double height, the Last row must be equal to 0x 10111 to display the status row in the two half pages.

ON SCREEN DISPLAY (Cont'd)

Figure 82. Memory Management for Scrolling Window



ON SCREEN DISPLAY (Cont'd)

DISPLAY CONTROL MODE 0 REGISTER (DCM0R)
R250 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
DE	STE	FRE	CE	GFR	GRE	SF	S/D

Bit 7 = **DE**: *Display Enable control bit.*
If DE is reset, no display will be performed. If DE is set, a display will be done as defined through the various control bits.

Bit 6 = **STE**: *Semi-Transparent Enable bit.*
This bit is active only in Single page display mode.

While the Display is disabled, the horizontal and vertical sequencers are forced in their reset state and the RGB & FB DACs are not off (still presenting on-chip resistors to Ground).

Note: This mode shows a visible grid on the screen.

Bit 5 = **FRE**: *Fringe Enable control bit.*
If this bit is set, and the SWE bit is reset (refer to the National Character Set Register description) the serial attribute 1Bh has a fringe toggle function.

FRE	SWE	1Bh Serial attribute acts as:
0	0	No Action
0	1	G0 Toggle
1	0	Fringe Toggle
1	1	G0 Toggle

Bit 4 = **CE**: *Conceal Enable control bit.*
0: Reveal any text defined as concealed by serial attributes (Default)
1: Conceal any text defined as concealed by serial attributes

Bit 3 = **GFR**: *Global Fringe Enable control bit.*
If this bit is set, the whole display is in fringe mode (except if a Double page display mode is programmed).

Bit 2 = **GRE**: *Global Rounding Enable control bit.*
If this bit is set, the whole display is in rounding mode (except if a Double page display mode is programmed).

Bit 1 = **SF**: *Screen Format control bit.*
0: Configures the Display for 4/3 TV screen format.
1: A fixed offset of 128 Pixel clock periods is added before any character is displayed; the Full Screen Color attribute is used while the offset is running.

The SF bit intended for displaying on 16/9 TV screen format tubes, the display picture will be re-centered.

Bit 0 = **S/D:** *Single or Double page control bit.*
0: A single page is displayed on screen (i.e. 40-character width).
1: A set of two pages is displayed contiguously (i.e. 80-character width).

Note: In 80 characters per row and in full page mode, line 25 of each field is displayed as a transparant line (as this line is not in the visible part of the screen, this should not present a limitation).

Programming a Double page display will automatically mask the Fringe, Rounding and Parallel Mode control bits. Their register values are not changed and they will automatically recover their initial effect if the display is switched back in a Single page mode.

ON SCREEN DISPLAY (Cont'd)

DISPLAY CONTROL MODE 1 REGISTER (DCM1R)
R251 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	EXTF	FBL	PM	SPM

Bits 7:4 = Reserved bits, keep in reset state.

Bit 3 = **EXTF**: *External Font.*
Only when the emulator is used, this bit selects the font memory containing a user-defined OSD font. In normal user application, this bit has no effect.
0: Internal font memory of the emulator chip.
1: External font RAM of the emulator board.

Bit 2 = **FBL**: *Fast Blanking Active Level control bit.*
The FBL bit must be reset if the on-screen display is done while the FB output is low. The FBL bit must be set if the on-screen display is done while the FB output is high. This bit also controls the TSLU AF output polarity with the same rule as for FB.

Bit 1 = **PM**: *Line Mode control bit.*
If PM is reset, the display is working in Full page mode, i.e. the screen is composed of one header, 23 text rows plus 2 status rows. If PM is set, the display works in Line mode.

Line mode allows up to 12 rows to be displayed anywhere on the screen. The row attribute (see TDSRAM mapping) contains the row numbers on the screen. The byte position of the row attribute conrresponds to the row in the TDSRAM. For example, if the 3rd byte of the row attribute contains 6, the 3rd row in TDSRAM will be displayed as the 6th row on the screen.

Bit 0 = **SPM**: *Serial/Parallel Mode control bit.*
If the SPM bit is reset, the display is done in Serial mode, i.e. a character or attribute is coded with a single byte. If the SPM bit is set, the display is done in Parallel mode, i.e. a character or an attribute is coded on two bytes.

TDSRAM POINTER REGISTER (TDPR)
R252 - Read/Write
Register Page: 32
Reset Value: 0000 0000 (00h)

7							0
HS3	HS2	HS1	HS0	PG3	PG2	PG1	PG0

Bits 7:4 = **HS[3:0]:** Location of the current Header and Status Rows in the TDSRAM.

Bits 3:0 = **PG[3:0]**: Location of the current Page content (rows 1 to 23) in the TDSRAM. For more details, refer to Section , ON SCREEN DISPLAY (Cont'd).

The HS[3:0] and PG[3:0] bits described by the R246 and R247 registers in page 32. Display locations, Head/Stat location, Page location, are physically the same: these sets of address bits can be modified through two different programming accesses.

ON SCREEN DISPLAY (Cont'd)

DISPLAY ENABLE 0 CONTROL REGISTER (DE0R)
R253 -Read/Write
Register Page: 32
Reset Value: 1111 1111 (FFh)

7							0
R8	R7	R6	R5	R4	R3	R2	R1

Bits 7:0 = **R[8:1]**: *Row display enable control bit*. When the "Ri" bit is set (Reset value), the corresponding row (with row in the page, numbered from 1 to 23) will be displayed. When the "Ri" bit is reset, the full screen color is displayed.

DISPLAY ENABLE 1 CONTROL REGISTER (DE1R)
R254 -Read/Write
Register Page: 32
Reset Value: 1111 1111 (FFh)

7							0
R16	R15	R14	R13	R12	R11	R10	R9

Bits 7:0 = **R[16:9]**: *Row display enable control bit*. When the "Ri" bit is set (Reset value), the corresponding row (with row in the page, numbered from 1 to 23) will be displayed. When the "Ri" bit is reset, the full screen color is displayed.

DISPLAY ENABLE 2 CONTROL REGISTER (DE2R)
R255 -Read/Write
Register Page: 32
Reset Value: x111 1111 (xFh)

7							0
x	R23	R22	R21	R20	R19	R18	R17

Bit 7 = Reserved

.

Bits 6:0 = **R[23:17]**: *Row display enable control bit*. When the "Ri" bit is set (Reset value), the corresponding row (with row in the page, numbered from 1 to 23) will be displayed. When the "Ri" bit is reset, the full screen color is displayed.

ON SCREEN DISPLAY (Cont'd)

DEFAULT COLOR REGISTER (DCR)
R240 - Read/Write
Register Page: 33
Reset Value: 0111 0000 (70h)

7							0
DFG3	DFG2	DFG1	DFG0	DBG3	DBG2	DBG1	DBG0

Bits 7:4 = **DFG[3:0]**: *Default Foreground Color.*
DFG[3:0] = (Half-Intensity, R, G, B)

Bits 3:0 = **DBG[3:0]**: *Default Background Color*
DBG[3:0] = (Half-Intensity, R, G, B)

Table of Color Values (hex)

0 Black	8 Black
1 Blue	9 Dark blue
2 Green	A Dark green
3 Cyan	B Dark cyan
4 Red	C Dark red
5 Magenta	D Dark magenta
6 Yellow	E Dark yellow
7 White	F Grey

CURSOR ABSOLUTE VERTICAL POSITION REGISTER (CAPVR)
R241 - Read/Write
Register Page: 33
Reset Value: 0000 0000 (00h)

7							0
0	0	0	ACP4	ACP3	ACP2	ACP1	ACP0

Bits 7:5 = Reserved, keep in reset state.

Bits 4:0 = **ACP[4:0]:** Absolute Vertical Position of the cursor in case of double height or scrolling.

TDSRAM PAGE POINTER REGISTER (TDPPR)
R246 - Read/Write
Register Page: 33
Reset Value: xxx0 0000 (x0h)

7							0
x	x	x	0	PG3	PG2	PG1	PG0

Bits 7:4 = Reserved, keep in reset state.

Bits 3:0 = **PG[3:0]**: *Page Pointer*
Location of the current Page content (rows 1 to 23) in the TDSRAM. For more details, refer to the Display Memory Mapping Section , ON SCREEN DISPLAY (Cont'd).

TDSRAM HEADER/STATUS POINTER REGISTER (TDHSPR)
R247 - Read/Write
Register Page: 33
Reset Value: xxx0 0000 (x0h)

7							0
x	x	x	0	HS3	HS2	HS1	HS0

Bits 7:4 = Reserved, keep in reset state.

Bits 3:0 = **HS[3:0]**: *Header/Status Rows Pointer*
Location of the current Header and Status Rows in the TDSRAM. For more details, refer to the Display Memory Mapping paragraph Section , ON SCREEN DISPLAY (Cont'd).

7.4.10 Application Software Examples

Before starting an OSD Display, it is very important to start all the internal clock/timings

To understand the software routines given below, make a thorough study of the chapters on the Reset and Clock Control Unit (RCCU) and the TDSRAM Interface.

Initialization of the Internal Clock

```
;=====================================================================
;           MAIN CLOCK INIT
;=====================================================================
CLOCKS::
;--------- CPU MAIN CLOCK ------------
                  ; C K M A I N  provided by the freq. multplier
            spp #TCCR_PG; Timings & clock Controller registers page
                  ; page 39 or 27
            ld MCCR,#0x05; program the frequency multiplier down
                  ; counter in the feed-back loop (253 =FD)
                  ; freq=(5+1)*2 =12Mhz
                  ; freq=(7+1)*2 =16Mhz
                  ; freq=(8+1)*2 =18Mhz
            ld MCCR,#0x85; enable the freq. multiplier
            srp #BK20
            ldw rr0,#0x2FFF;
time_stab1:       ; for frequency multiplier stabilization
            decw rr0; change CPU source clock & wait clock stabilization
            cpw rr0,#0x00;
            jxnz time_stab1;
            ld MCCR,#0xC5 ; select the freq. multiplier as main clock
            pop ppr
;=====================================================================
;           SYNCHRO START
;=====================================================================
            spp #SYCR_PG   ; set page pointer to page 23h or 35 decimal
            ld CSYCTR,#000h; R243, Hsync and Vsync from deflection part
(external)
            ld CSYSUR,#0C4h; R242 Sync Controller Set-up register
                  ; Standard mode, Positive polarity of Hsync & Vsync
                  ; delay on Hsync/Vsync HSF(3:0)=9
;=====================================================================
;           DISPLAY PIXEL CLOCK
;=====================================================================
;
            spp #TCCR_PG; Timings & clock Controller registers page
                  ; set page pointer to page 39 decimal
            ld SKCCR, #0x09; FE, Skew clock control register
                  ; program the frequency multiplier down
                  ; counter in the feed-back loop
```

```
                ; dot_freq= 4Mhz(4+1)=20Mhz (4/3)
                ; dot_freq= 4Mhz(5+1)=24Mhz (16/9)
                ; divide by 2
            ld SKCCR, #0x89; enable the freq. multiplier
            srp #BK20
            ldw rr0,#0x0FFF;
time_stab2:     ; for frequency multiplier stabilization
            decw rr0; SKEW clock stabilization
            cpw rr0,#0x00 ;
            jxnz time_stab2;
            ld PXCCR,#0x80;(PXCCR)  start Pixel Line PLL
            spp #TDSR_PG2; page 26h, TDSRAM Controller registers third page
            srp #000h
            ld CONFIG, #003h; FC, ram Interface Configuration register
                ; enable display and Dram access
;=====================================================================
```

Initialization of the OSD in Serial Mode

```
;=====================================================================
;  INIT DISPLAY ROUTINE
;=====================================================================
INIT::
;-----------Display Position & Black Reference
            spp #DMP1_PG; page 020h Display memory map registers page
            ld HBLANKR,#0x45; HBLANKR register [7: 0]; reset=03
                ; important delay for black reference on RGB cathod
            ld HPOSR,#0x35; HPOSR register [7:0]; reset=03
            ld VPOSR,#0x10; VPOSR register [5:0]; reset=00
;-----------------
            spp #DMP1_PG; page 020h Display memory map registers page
                ; F3, Full Screen Color register
            ld FSCCR, #0x01  ; no subtitle mode
                ; BE, Box enable
                ; TIO, Text in/out
                ; MM, Mixed Mode
                ; FSC[3:0]=half blue full screen
            ld HSCR, #03Fh; bit5,   4,   3,   2,  1,  0
                ; ES1 , NS1, ES0, NS0, EH, NH
                ; 0x3F > set header and status in level 1+
                ;        (parallel)
                ; 0x2a > set header and status in level 1
                ; F5, National Characters register
            ld NC, #010h; SWE, NCM, NC[3:0]
;------- Scrolling INIT ----------
            spp #DMP1_PG; page 020h Display memory map registers page
```

```
                ; F8, Scrolling Control Line register
            ld SCLR ,#000h ; SCE, FSC, SS, FIRSTROWSCRO[4:0]
                ; F9, Scrolling Control Horizontal register
            ld SCHR ,#02fh ; DH, ER, UP/D, LASTROWSCRO[4:0]
;------- Cursor position
                ; F8, Scrolling Control Line register
            ld SCLR ,#000h ; SCE, FSC, SS, FIRSTROWSCRO[4:0]
                ; F9, Scrolling Control Horizontal register
            ld SCHR ,#02fh ; DH, ER, UP/D, LASTROWSCRO[4:0]
                ; F6, Cursor Horizontal Position register
            ld CHPOSR , #005h; CURSOR HPOS [6:0]
                ; F7, Cursor Vertical Position register
            ld CVPOSR , #000h; FON, CM[1:0], CURSOR VPOS[4:0]
;------- Control
                ; FA, Control Mode 0 register
            ld DCM0R,#0a0h; DE, STE, FRE, CE, GFR, GRE, SF=4/3, S/D=40
                ; display enable
                ; solid mode
                ; toggle fringe enable
            ld DCM1R,#0x04; register 251 (FBh) Control Mode 1 register
                ; DAT[6:4]/bits 7,6,5 & TDR/bit4
                ; FNEX=0, on-chip font
                ; FBL=1 fastblanking active high
                ; PM =0 Full page mode
                ; SPM =0 serial mode
;--------Dram location: header/status rows, current display
                ; FC, Dram Location register
            ld TDPR, #080h; HS[3:0], AD[3:0]
                ; for header bit12=1
;------- foreground/background
            spp #DMP2_PG; page 021h Display memory map registers page
            ld DC, #07Fh ; reg. F0h, DFG [3:0], DBG [3:0]
                ; FG full white
                ; BG grey (half white)
;-----------------------------
            spp #DMP1_PG; page 020h Display memory map registers page
            ld DE0R, #0FFh; ROWEN [8:1]
            ld DE1R, #0FFh; ROWEN [16:9]
            ld DE2R, #0FFh; ROWEN [23:17]
            ret
=========================================================================
```

7.5 SYNC CONTROLLER

The SYNC Controller receives Horizontal / Vertical sync information coming from the chassis. The VSYNC and HSYNC inputs use schmitt triggers to guarantee sufficient noise rejection.

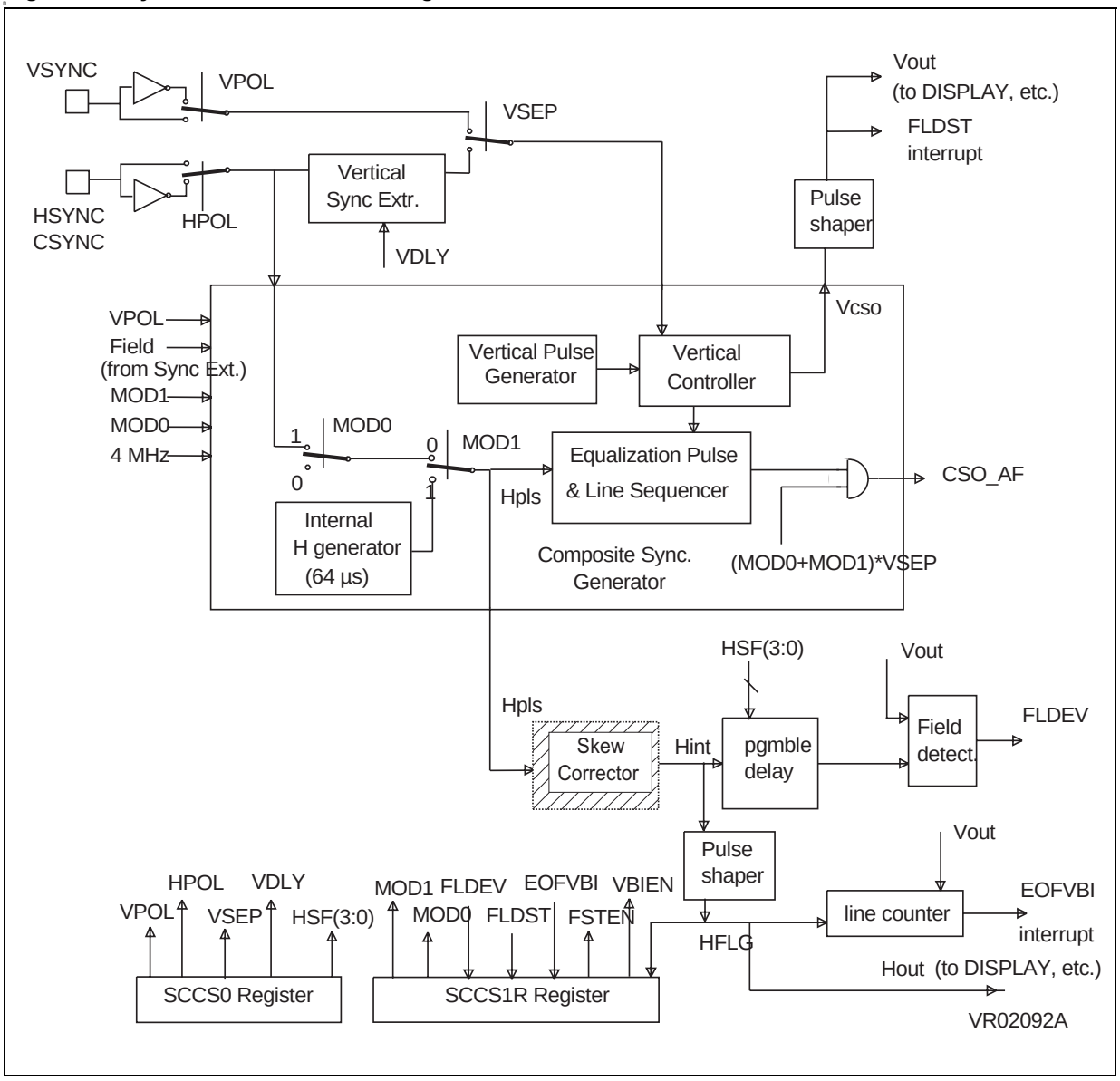
The SYNC Controller unit provides the H internal sync signal to the Display Skew Corrector, which rephases the Pixel clock.

It provides also the H and V internal sync signals to the TDSRAM Controller to perform correct TV line counting, thus generating the necessary time windows for a proper TDSRAM access real time sharing by the Teletext Acquisition, the Display Controller and the CPU (for more details refer to the TDSRAM Controller chapter). Field information is also available for the Display Controller.

The SYNC Controller unit also generates two interrupt sources corresponding respectively the TV field start and to the end of VBI event ("VBI" stands for Vertical Blank Interval).

The SYNC Controller implements also a "Composite Sync" signal generator which provides a composite sync output signal (called CSO) available through an I/O port alternate function.

Figure 83. Sync Controller Block Diagram



SYNC CONTROLLER (Cont'd)

7.5.1 H/V Polarity Control

Two control bits manage the H/V polarities. HPOL (SCCS0R.6) manages the HSYNC polarity (a positive polarity assumes the leading edge is the rising one). VPOL (SCCS0R.7) controls the VSYNC polarity.

7.5.2 Field Detection

Field detection is necessary information for the Display controller for fringe and rounding features.

To determine correctly the field in case of using separate H and V input signals, it is necessary to provide an internal compensation of the hardware delay generated on VSYNC (VSYNC is generally issued by integrating the equalization pulses). Therefore the VSYNC leading edge is compared to the leading edge of an internally delayed HSYNC.

The delay applied to HSYNC is software programmable through the SCCS0R (3:0) bits (from 0 to 63 µs). It must be calculated by the user as being the time constant (modulo 64 µs) used to extract VSYNC by the other components of the chassis.

7.5.3 Interrupt Generation

The SYNC Controller unit can provide two different interrupts to the ST9+ Core. The first interrupt appears at each beginning of field upon detection of the Vertical Sync pulse coming from the deflection circuitry (i.e. from VSYNC); it is called the "Field start" interrupt. A flag is associated to this interrupt, called "FLDST" (SCCS1R.6). This flag is set to "1" by hardware when the Vertical Sync pulse appears. It must be cleared by software.

The second interrupt appears at the end of each Vertical Blank Interval. It is generated at the beginning of the line 25 counted from the deflection circuitry (i.e. from VSYNC); and is called the "End OF VBI" interrupt. A flag is associated to this interrupt, called "EOFVBI" (SCCS1R.7). This flag is set to "1" by hardware when the line 25 starts. It must be cleared by software.

These two interrupts EOFVBI and FLDST are respectively attached to the INT4 and INT5 external interrupt inputs of the ST9+ Core. The leading edges of the 2 interrupt requests are the falling ones. (For more details, refer to the Interrupts chapter).

7.5.4 Sync Controller Working Modes

Different working modes are available fully controlled by software.

The first two working modes assume that TV deflection sync signals are available and stable.

The last two modes assume that no TV signal is available. The chip works in a free-running mode providing standard TV Sync signals based on the main internal 4 MHz clock.

Switching from one mode to any other is done under full software control, through the programming of two control bits called as MOD1 and MOD0. These control bits are described in the SCCS1R register

7.5.4.1 Standard Sync Input Mode

This mode is accessed when both MOD1 and MOD0 bits are reset.

In this mode, the µP receives the chassis synchro through two separate inputs. These are VSYNC and HSYNC. It also assumes the VSEP (SCCS0R.5) is reset.

SYNC CONTROLLER (Cont'd)

7.5.4.2 Composite Sync Input Mode

This mode is very similar to the "Standard Sync Input Mode" described above. It is also accessed when both MOD1 and MOD0 bits are reset.

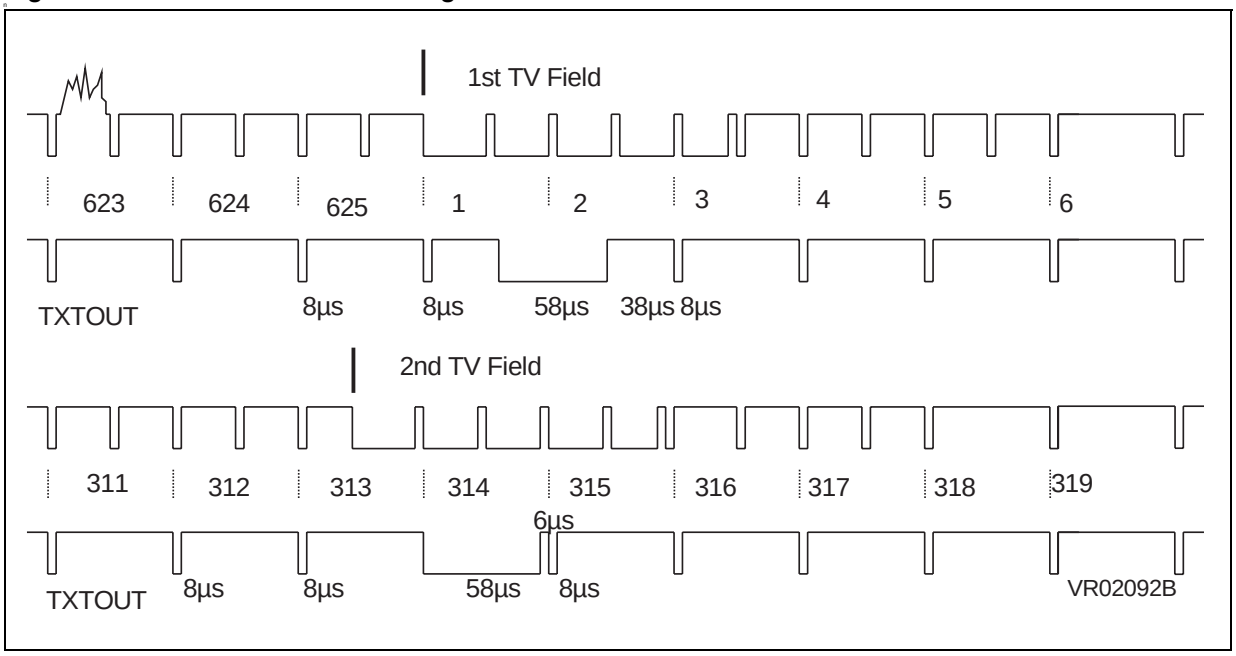
In Composite Sync mode, a single CSYNC/HSYNC input pin is used to enter both the horizontal and vertical sync pulses (VSEP control bit is set to 1). In this mode, the VSYNC pin must be tied to VSS on the application board to prevent a floating CMOS input configuration.

The CSYNC signal characteristics are assumed to perfectly respect the STV2160 TXTOUT pin specification which is reviewed in Figure 2 & Figure 3.

The vertical sync signal is extracted from the CSYNC signal by the mean of an Up/Down counter used as a digital integrator. The counter works in "Up" mode during the sync pulses.

Two time constants can be programmed using the VDLY control bit (refer to the register description). The smallest one corresponds to 16µs; the second one being 32µs.

Figure 84. STV2160 TXTOUT Timings



SYNC CONTROLLER (Cont'd)

7.5.4.3 Free-Running Monitor Sync Mode

This mode is accessed when the MOD1 bit is set.

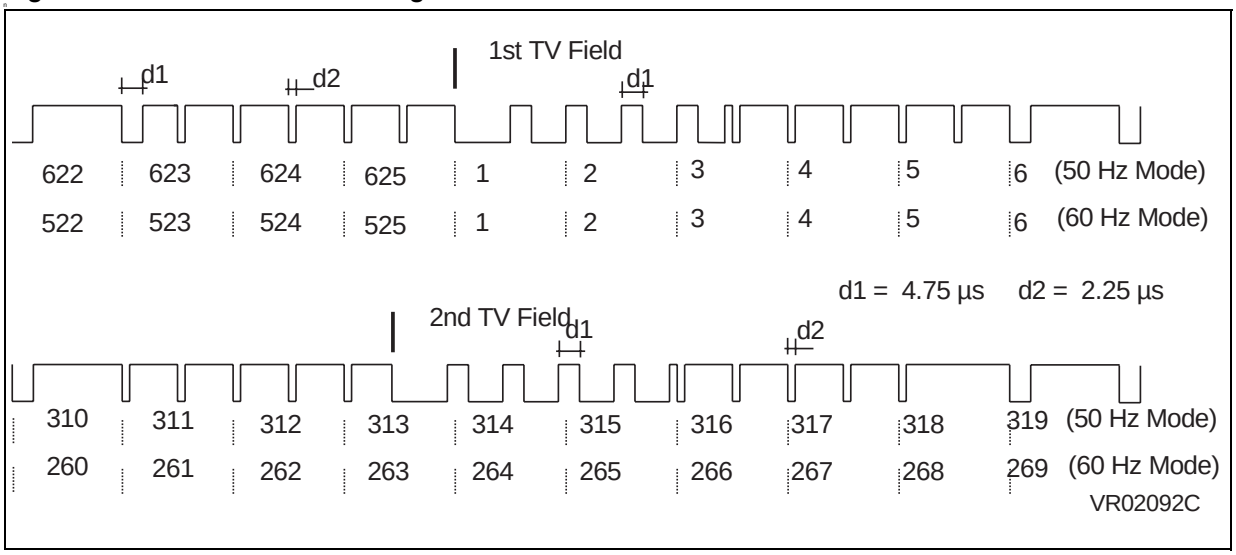
In this mode, the chassis HSYNC and VSYNC signals are not used. They are replaced by the sync signals which are fully Crystal based (use of the internal main 4 MHz Clock).

Two free-running monitor modes are available: when the MOD0 bit is reset the Composite Sync output (CSO) is generated for a 60Hz format; when the MOD0 bit is set to "1" the Composite Sync output (CSO) is generated for a 50Hz format. For both formats, the TV line period is 64µs.

The Composite Sync alternate function Output (CSO) can be activated or disabled under control of the VSEP bit.

In Free-Running Monitor Sync mode, the VPOL control bit is used to control whether an interlaced or non-interlaced TV context must be generated. When the non-interlaced context is programmed, only the "1st TV Field" configuration is generated.

Figure 85. Even/Odd Field Timings



SYNC CONTROLLER (Cont'd)

7.5.5 Register Description

SYNC CONTROLLER CONTROL AND STATUS REGISTER 0 (SCCS0R)
R242 - Read/Write
Register Page: 35
Reset value: 0000 0000 (00h)

7							0
VPOL	HPOL	VSEP	VDLY	HSF3	HSF2	HSF1	HSF0

Bit 7= **VPOL**. *VSYNC Polarity*
When MOD[1:0] are reset, this bit configures the polarity of the VSYNC input.
0: Negative polarity (leading edge is falling edge)
1: Positive polarity (leading edge is rising edge)

For other cases:

MOD1	MOD0	VPOL	
0	0	x	VSYNC polarity control
x	1	0	interlaced
x	1	1	non-interlaced
1	x	0	interlaced
1	x	1	non-interlaced

Bit 6= **HPOL**. *HSYNC/CSYNC Polarity*.
0: Negative polarity (leading edge is falling edge)
1: Positive polarity (leading edge is rising edge)

Bit 5= **VSEP**. *Separate Sync*
This bit selects either HSYNC/VSYNC or CSYNC synchro controller mode. It is only valid when MOD[1:0] are reset:
0: The standard mode using two inputs (VSYNC and HSYNC) is activated.
1: The Composite Sync mode is activated; the CSYNC input (on the same pin as HSYNC) will be used to get both H and V signals.

For other cases:

MOD1	MOD0	VSEP	CSO alternate function
0	0	x	disabled
x	1	0	disabled
x	1	1	enabled
1	x	0	disabled
1	x	1	enabled

Bit 4= **VDLY**. *Vertical Delay control bit*.
This bit is active only if the Composite Sync mode is enabled. The selection of this bit can effect noise margin (longer delay is better) and the field detection.
0: Vertical is generated after detecting a pulse greater than 16µs
1: Vertical is generated after detecting a pulse greater than 32µs

Bits 3:0= **HSF**. *Horizontal Shift for Field detection*.
These 4 bits program the delay, in steps of 4µs (50Hz) or 2µs (100Hz), applied to the HSYNC pulse in order to properly determine the field information by comparison with VSYNC. This value is a chassis constant depending upon the way the separate H/V signals are generated.

SYNC CONTROLLER (Cont'd)

SYNC CONTROLLER CONTROL AND STATUS REGISTER 1 (SCCS1R)
R243 - Read/Write
Register Page: 35
Reset value: 0000 0000 (00h)

7							0
EOFVBI	FLDST	FLDEV	HFLG	FSTEN	VBIEN	MOD1	MOD0

Bit 7= **EOFVBI:** *End Of VBI Flag*.
This bit is set to "1" by hardware at the beginning of the line 25 of the current field, when the End of VBI interrupt request is sent to the Core. The EOFVBI flag must be reset by software before the end of the current field.

Bit 6= **FLDST:** *Field Start Flag*.
This bit is set to "1" by hardware on the leading edge of the vertical sync pulse when the field start interrupt request is forwarded to the Core. The FLDST flag must be reset by software before the end of the current field.

Bit 5= **FLDEV:** *Field Even Flag*.
This bit is read-only. It indicates which field is currently running;
0: First field is running
1: Second field is running

Bit 4= **HFLG:** *Horizontal Sync Flag*.
This bit is read-only. It just copies the Horizontal sync transient information issued by the horizontal pulse shape unit. The bit is read at "1" at during each H sync pulse and lasts to "1" up to 4 µs.

Bit 3= **FSTEN:** *Field Start Interrupt Enable*.
0: The FLDST interrupt is disabled and the external interrupt pin becomes the interrupt input.
1: The FLDST interrupt is enabled and the interrupt from the external pin is disabled.

Bit 2= **VBIEN:** *VBI Interrupt Enable*.
0: The EOFVBI interrupt is disabled and the external interrupt pin becomes the interrupt input.
1: The EOFVBI interrupt is enabled and the interrupt from the external pin is disabled.

Bits 1:0= **MOD[1:0]**

MOD1	MOD0	H & V sync sources	CSO	CSO generator
0	0	chassis sync pulses	no	-
0	1	reserved		reserved
1	0	from Xtal (60 Hz)	yes	on-chip timing generator; free-running
1	1	from Xtal (50 Hz)	yes	on-chip timing generator; free-running

7.6 SYNC EXTRACTOR

The Sync Extractor extracts the Sync pulses from the CVBS signal and determines the CVBS HSYNC and VSYNC signals. The CVBS HSYNC signal is fed to the VPS/WSS slicer and to the Teletext Slicing Unit to re-synchronize them at each TV line.

The Sync Extractor Unit provides the CVBS VSYNC pulse to the Csync Controller unit which might use it for some standalone modes when the chassis time base is switched off (refer to the Sync Controller chapter for more details).

7.6.1 Time Windowing For Slicers

A line counter counts the CVBS HSYNC pulses in order to open and close the proper windows within each field for slicing the Teletext (lines 6 to 22 for the first field, lines 318 to 335 for the second field), VPS (lines 15 to 17) or WSS (lines 22 to 24) data. The line counter is able to count a complete field before overflowing.

An extended Teletext Search window can be programmed by software through the TXTES bit of the Teletext Slicer Control register (TXSCR) (refer to the Teletext Slicing Unit chapter). The extended window starts at line 2 (first field) or line 315 (second field).

7.6.2 Field Detection

To secure the slicing units functionality, a field detection is implemented. This field detection flag is updated at each beginning of field. In case a field start is missed, the field detection flag will keep toggling according to its previous value and the end of field predicted by the line counter (312 or 313 lines).

7.6.3 CVBS Amplitude Information

The Sync Extractor unit provides information about the real time amplitude of the CVBS signal. This information is obtained by storing the peak value of the sync pulse referred to the black level. This analog voltage is bufferized and forwarded to the on-chip A/D Converter.

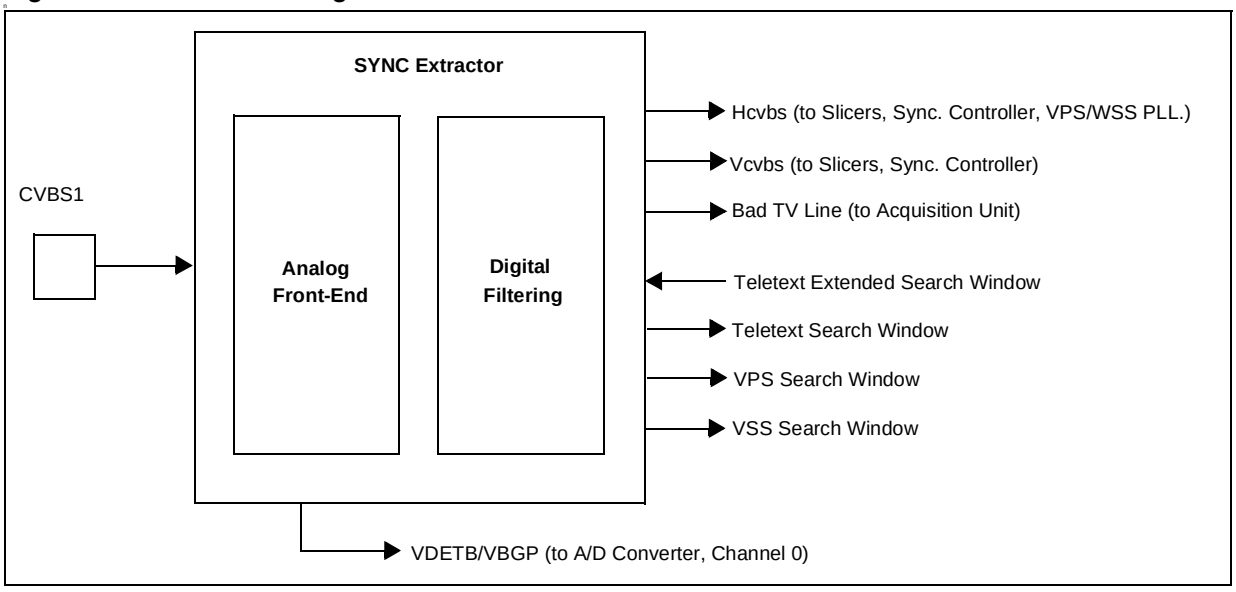
7.6.4 CVBS Signal Quality

The quality of the CVBS Sync pulses is also checked by the Sync Extractor according to the following criteria: a TV line is declared “bad” if two or more HSYNC pulses are found within 64 µs, or if the HSYNC pulse is missing. Each time a “bad” TV line is detected, a “Bad TV line” pulse is forwarded to the Acquisition which is part of the Teletext Slicing Unit. These pulses are counted only during the Vertical Blank Interval, providing a quality criterion of the CVBS signal at the time the slicers treat it. For more details on the subject, refer to the Teletext Slicing.

7.6.5 Sync Extractor Switch Off

When the VPS/WSS and Teletext slicers are switched off, the Sync Extractor automatically enters an idle state to prevent current consumption.

Figure 86. Time Windowing for Slicers



7.7 TELETEXT SLICER AND ACQUISITION UNIT

7.7.1 Introduction

The Teletext Slicer processes the incoming Composite Video signal (CVBS1 pin) to extract Teletext data.

The Acquisition Unit filters the data flow from the slicer and transfers valid line data to the acquisition buffer where it is read by the TDSRAM interface.

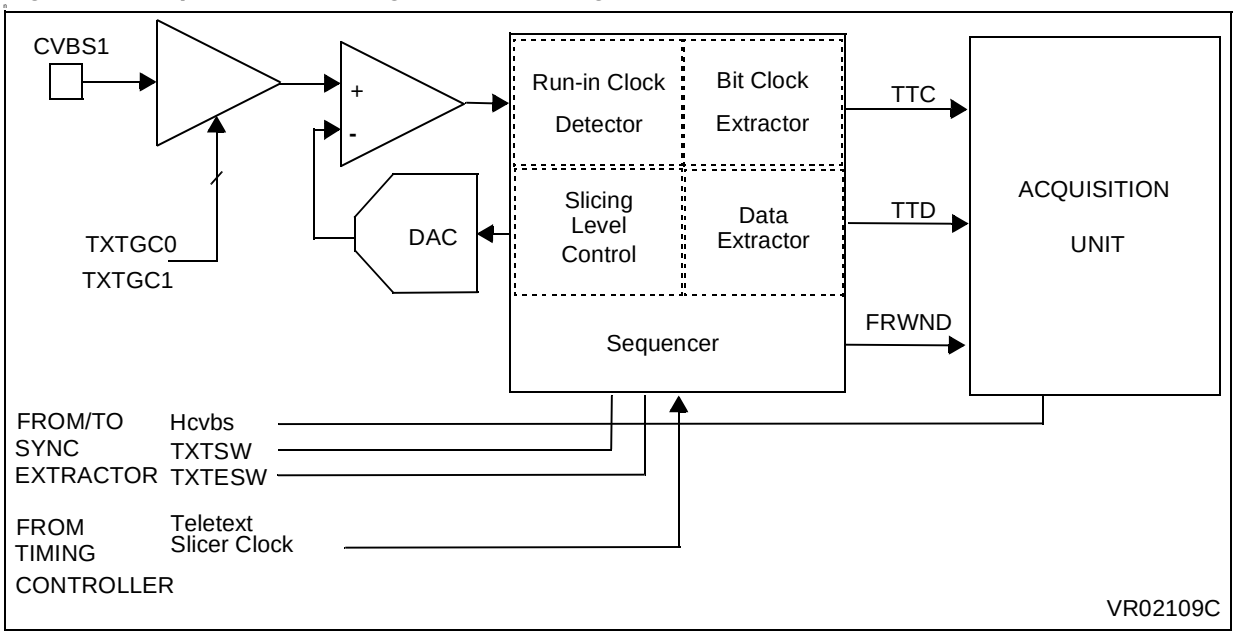
A Hamming decoding function is available for checking or correcting data after storage.

The Acquisition unit also provides a Teletext signal quality measure function to evaluate the quality of the Teletext data reception by testing the HSYNC pulses or the Hamming error numbers during the VBI.

7.7.2 Main Features

- Hardware framing code check
- Hardware & software data prefiltering per magazine
- Real time hardware Hamming 8/4 check on incoming data
- Raw data storage of packets in physical order
- Fully programmable storage address
- 2 acquisition paths for magazine pages and service packets
- Asynchronization between deflection and CVBS Syncs.
- CVBS signal quality measure through a "Bad TV line" counter
- Hamming 24/18 or 8/4 one-shot decoding.

Figure 87. Acquisition & Slicing Unit Block Diagram



TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.3 Teletext Slicer

The CVBS1 input voltage is 1Vpp +/- 6dB (i.e. from 0.5Vpp to 2.0Vpp). The CVBS1 pin is AC driven through a 470 nF capacitor.

Note: An indicator of Teletext Slicer quality criterion is to have data error-free (or less than 3 errors per page) for an **eye height** reduction down to **30%**.

7.7.3.1 CVBS Amplitude Control

A video pre-amplifier treats the CVBS internal signal. Software should adjust the signal amplitude for the most effective working range of the Data Slicer while keeping an optimized signal/noise ratio. The pre-amplifier gain is programmed through two bits in the TXSCR register.

The composite video signal enters the Sync Extractor through the CVBS1 pin. The sync pulse's peak value (with reference to black level) is stored. This signal's pulse amplitude (VDETB) is then fed to channel 0 of the A/D converter. The conversion result can then be used for programming the best gain value of the video pre-amplifier. See Figure 87.

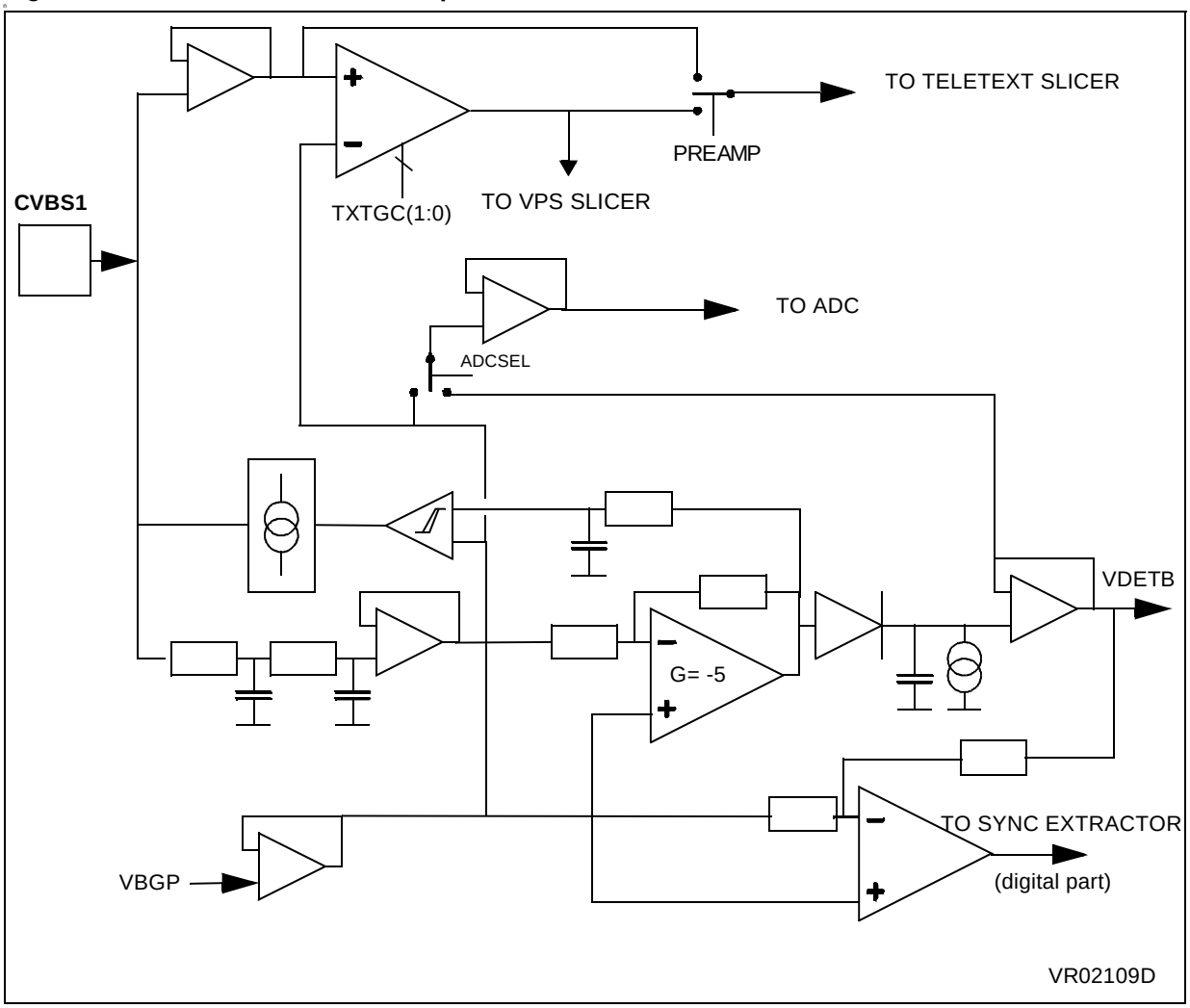
The theoretical gain calculation is the following:

– Assume Atext is the effective peak amplitude of the Teletext data within the video signal (this peak amplitude being referred to the black level clamp),

– The effective working range of the Data Slicer is equal to 1.2V

– Theoretical gain= 1.2/ Atext

Figure 88. CVBS1 Functional Description



TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.3.2 Signal Windows

TXTEN (TXSCR.7) is the Teletext Enable bit. Setting this bit allows the slicer to check for the presence of Teletext data. The search for data is from line 6 to line 22 for field 1 and from line 318 to line 335 for field 2. The search window's timing is defined by the signal generated by the Sync Extractor. Information from this signal is found on the bit TXTSW (Teletext Search Window, TXSCR.3). An extended search can be done by enabling TXTES (Teletext Extended Search, TXSCR.6). When this bit is enabled, the search in field 1 will start on line 2 and in field 2 on line 315.

During each search interval's TV line the slicer looks for the presence of the Run-In-Clock. When the Run-In-Clock is found, data extraction is performed and the Teletext Data (TTD) and Clock (TTC) are sent to the Acquisition Unit. Enabling the TTC clock is done using the SLCCR.7 bit. Refer to the Timing and Clock Controller Chapter.

If the Run-In-Clock is not found within 12 µsecs. from the horizontal synchro leading edge, all data processing is stopped and no FRWND (Framing Code Window) is generated. See Figure 88.

The Teletext Slicer generates the FRWND control signal. This signal defines when the Acquisition Unit must check for the presence and validity of the framing code.

The FRWND is activated at the end of the running clock and it lasts 24 teletext data bits.

The Slicer uses the Teletext slicer clock generated by the Timing Controller. This clock must be enabled before enabling the Slicer to guarantee correct behaviour. To synchronize all the control signals on a TV line base, the Horizontal Sync (Hcvbs) from the Sync Extractor is used.

7.7.4 Acquisition Unit

When Teletext data are available, they are broadcast during the Video blank interval (VBI).

During the Teletext window, the Teletext Slicer sends to the Acquisition Unit:

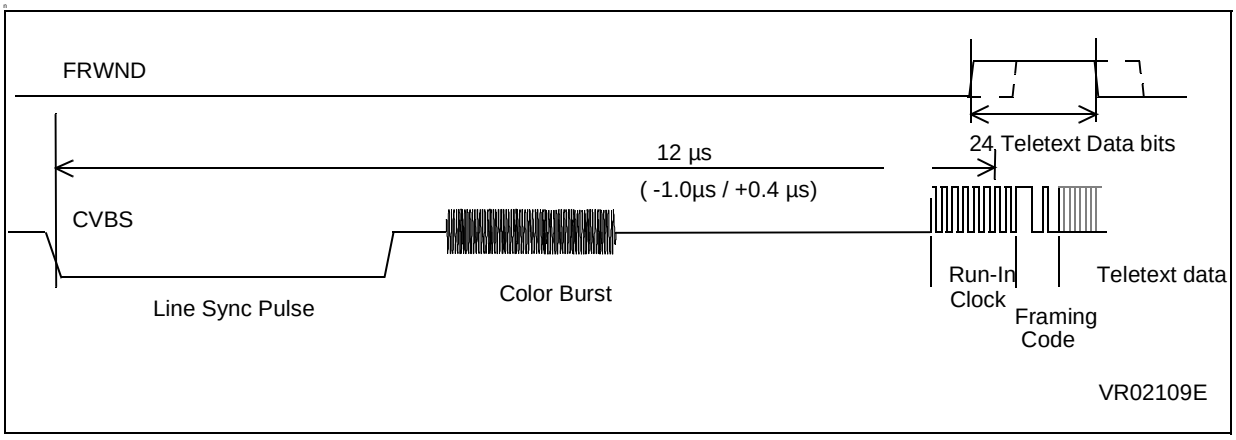
– TTC clock (at 6.9375 MHz)

– TTD serial data flow

– FRWND signal which enables the framing code search within a precise window for each TV line

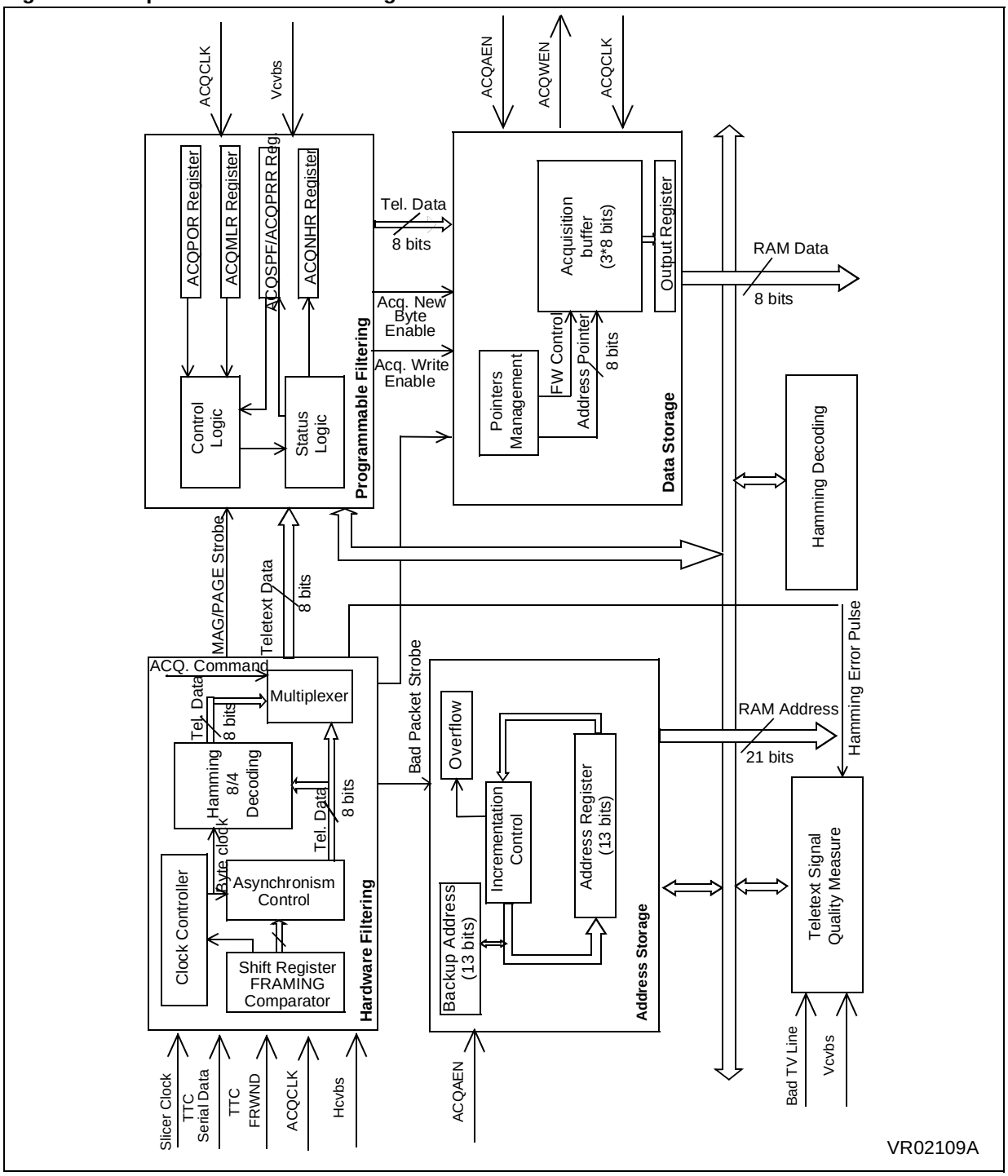
This FRWND signal is provided to the Acquisition Unit only during the TV lines that may contain Teletext data.

Figure 89. Framing Window Timing (FRWND)



TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

Figure 90. Acquisition Unit Block Diagram



TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.4.1 Data Extraction and Synchronization

Teletext data extraction is based on synchronization signals from the CVBS1 input video composite signals (Hcvbs and Vcvbs).

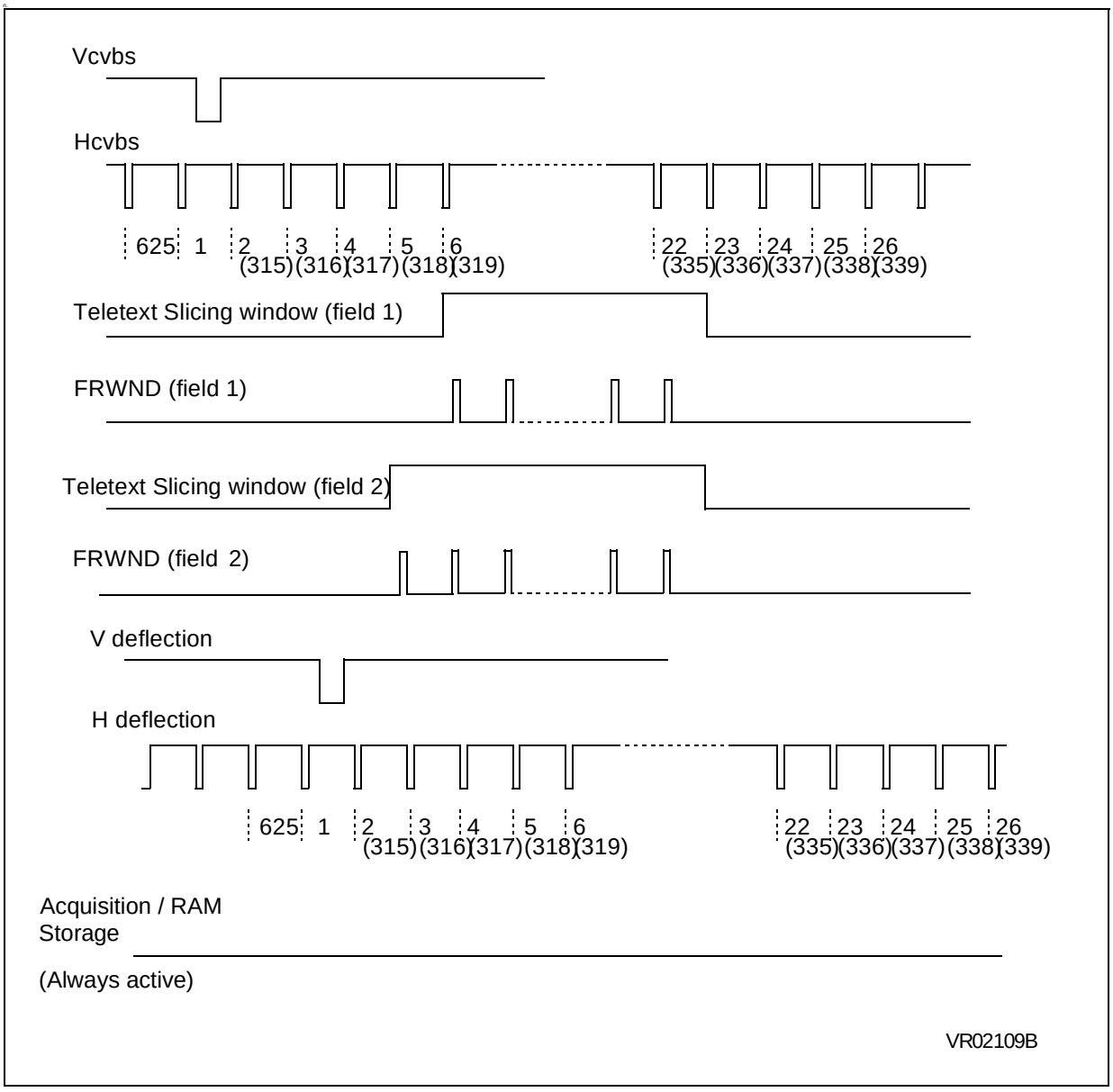
Teletext data storage into the internal TDSRAM is synchronized with the sync signals, HSYNC/ VSYNC, delivered by the chassis and also called "deflection sync signals". The storage of the data can be done during the whole field.

In practice, a significant phase shift of a few TV lines may appear between the deflection sync signals and the CVBS ones.

A Teletext data acquisition window is opened on line 6 of field 1 and line 317 of field 2 from the CVBS signal (refer to Figure 90).

The Teletext data storage is done independently of the phase shift.

Figure 91. Deflection / CVBS Timing



TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

Data Filtering

The Acquisition Slicer's hardware validates the correctness of the data to be filtered. A second filter called "programmable filtering" allows software to authorize or disable any data storage.

Hardware Filtering

When the Acquisition Unit receives the gated FRWND signal, it will search for the Framing code sequence in this time window.

If no valid framing code is detected, the Acquisition Unit waits for the next TV line (i.e. the next FRWND signal), ignoring any serial data of the current line.

If a valid framing code is detected, the Acquisition Unit starts a serial to parallel data conversion synchronized by the framing code detection, storing in byte format Teletext data of the current line.

After a valid Framing code, the first two bytes are Hamming 8/4 checked. This check allows up to one error to be automatically corrected. In this case the corresponding byte is assumed to be error free. These two bytes contain the magazine and the packet numbers.

If uncorrectable Hamming errors are found, the whole TV line data flow is rejected until the next valid Framing code is received.

If no Hamming error is found, these 2 bytes, magazine and packet numbers, are concatenated in a single byte, the 3 MSB being the magazine value. This concatenated byte is then filtered without any further hardware check, leading to a total of 41 sampled bytes per TV line.

Programmable Filtering

The programmable filtering unit implements two different acquisition paths, activated or stopped under full software control.

The filtering decision (store or reject) is taken by hardware in real time, by checking the first byte value (concatenated magazine / packet). This decision is then kept until the end of the 41 bytes flow, generating a "write enable" flag which is forwarded to the Data Storage Unit.

The content of the Data Storage Unit is then transferred to a TDSRAM memory area called "VBI Area" (see also Section 7.7.4.3, Address Generation).

Service Packets

The first acquisition path is dedicated to the ancillary packets X/29, X/30 and X/31. Three control bits associated with each of these Service packets, enable (if set) or disabling (if reset) the storage of the corresponding Service packet whatever the associated magazine and the ACQPOR or ACQMLR value (refer to the "Packet Request" register description).

In case a service packet storage was requested, the whole 41 bytes TV line data flow is transferred (i.e. written) to the data storage area. If it was not requested, then the 41 bytes corresponding data flow will be rejected (i.e. not transferred to the data storage area).

A "service packet" flag, called "SPF", is available. This flag SPF must be reset by software before the beginning of the next VBI. This flag will be set by hardware each time a service packet is received and was requested. This flag is common to the three types of Service packets. Reading it to "1" indicates that at least one (or more) Service packet was received and stored during the previous VBI..

Pi (i= 29, 30, 31)	SPF	results
0	0	no storage even if Pi broadcasted
1	set to 1 if Pi received	storage of Pi, if received

Warning: Accessing the Packet Request Register during the VBI when the Teletext Acquisition is running may lead to an incorrect functioning of the filtering unit.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

Page Handler

The page handler controls all packet types other filtered than those already mentioned (i.e. packets X/0 to X/28).These packets are filtered magazine-wise by the Acquisition Unit.

Two registers, called "Page Opened" Register (ACQPOR) and "Magazine Locked" Register (ACQMLR), are programmed by software before each VBI and define for each magazine which filtering type is requested.

Note: A Teletext page is opened by receiving a "header" (a header corresponds to a packet X/0). A page can not be closed by hardware. Also, receiving a service packet does not close the current page of the corresponding magazine (if it was already opened).

Assuming magazine i (counted from 0 to 7), the associated bits PORi and MLRi act as follows:

MLRi	PORi	Comments
1	0	Magazine locked. No storage even a header coming
0	0	Page closed, wait for new header
0	1	Page opened. Storage

When a magazine is closed, then no Data of this magazine will be stored (except for the Service packets described in the preceding paragraph).

When a "page closed" configuration is set up, then no data of the corresponding magazine will be stored until the next header of that magazine is received. Starting from this new header, with a «page open» configuration, all the following magazine information will be stored till the end of the VBI. The data storage will be always done to the VBI area.

A status register called "New Header Received" register (ACQNHR) associated to these control registers, is also available. It provides for each magazine a flag NHRi which must be reset by software before the beginning of the next VBI. This flag will be set to "1" by hardware when a new header of the corresponding magazine is received and stored.

Warning: Accessing the "Page Opened", the "Magazine Locked" or the "New Header Received" Registers during the VBI when the Teletext Acquisition is running may lead to a malfunctioning of the filtering unit.

Once the VBI is finished, the application can easily determine what happened during the VBI by checking the current state of the 3 registers (ACQPOR, ACQMLR and ACQNHR). The following table clarifies the different possible situations that may be produced at the end of the VBI.

MLRi	PORi	NHRi	Comments
1	0	-	Magazine locked; no storage done
0	0	0	No storage for that magazine during the VBI
0	0	1	Storage started on new header received
0	1	0	Current page not yet closed; data received or not
0	1	1	Current page closed; new page started

Warning: The (MLRi=1 & PORi=1) combination is reserved for non-user mode.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.4.2 Data Storage

The data issued by the filtering unit is then sent to the data storage area, which forwards these bytes to the TDSRAM controller for storing in TDSRAM memory. As the TDSRAM interface is synchronized with the chassis sync signals, not necessarily in phase with the CVBS signal, a data stack is implemented in the data storage area. This data stack is called the "Acquisition Buffer".

The Acquisition Buffer is tailored to stack on a TV line base, all the Teletext data that may come while no RAM storage is possible. This leads to a 3 byte buffer (for more details, refer to the TDSRAM Interface Specification).

The data bytes issued by the data filtering unit are stored into the Acquisition Buffer only if they passed successfully the different checks (explained in the preceding chapters; i.e. if the "Write Enable" flag received from the filtering unit is set). If the "Write Enable" flag is "0" then the current byte will be rejected (not written into the Acquisition Buffer).

Acquisition Buffer Management

After being written to by the filtering unit, the Acquisition Buffer is read by the TDSRAM interface to transfer data to the internal TDSRAM. As the two accesses are not "in phase", the Acquisition Buffer is built as a FIFO controlled by two pointers: write and read. Each time a data transfer to the TDSRAM interface is performed, a control line called "ACQWEN" will tell the TDSRAM interface if a write to the TDSRAM must be performed.

The ACQWEN line is set each time data is available in the buffer (read and write pointers comparison). This allows a TDSRAM write. If the buffer is empty read and write pointers comparison), the ACQWEN line is reset and no TDSRAM write is allowed. The TDSRAM interface will perform a read operation to the TDSRAM; for more details, refer to the TDSRAM interface specification.

7.7.4.3 Address Generation

The address generation unit provides to the TDSRAM interface the real time address where the byte must be stored into the internal TDSRAM. The address is generated on a 14 bits word length, allowing to access the TDSRAM memory as a contiguous space of up to 16 K-bytes.

The TDSRAM memory Address, referred as "ADDREG", is software programmable through 2 registers ACQAD1R, ACQAD0R (refer to the registers description for more details).

The ADDREG must be initialized by software. Afterwards during each VBI, it will be incremented each time a byte is stored in the corresponding memory area. The ADDREG follows a post-increment scheme, i.e. it is always pointing to the next free byte location.

A second register, called the "Backup Address" register, is implemented for storing the TV line packet (under reception) address's first byte. An HCVBS (Horizontal Sync Pulse) could be detected before the end of the TV line Teletext data is received because a sync pulse arrived during the time when Teletext data was sliced, so the TV line looks too short. The Backup Address value is reloaded into the ADDREG, providing the erasure of the already stored data of the "too short" TV line. At the same time, the read and write pointers of the Buffer are re-initialized to the first Buffer location (empty Buffer status).

Note: A TV line which is longer than 41 bytes is considered "too long". Only the first 41 bytes of data are stored. The next acquisition storage starts after a valid Framing code detection.

Warning: Accessing the ADDREG Registers during the Vertical Blank Interval when the Teletext Acquisition is running may lead to a malfunctioning of the TDSRAM interface addressing.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.5 Hamming Decoding

The Hamming Decoding Unit is a separate block that the application software can use to check and decode Hamming coded data groups of three bytes (Hamming 24/18 format) or one data byte (Hamming 8/4 format).

The bytes must either be written to the three corresponding registers ACQHD2R, ACQHD1R and ACQHD0R or only to ACQHD0R. The bytes must be written in the correct order, where the last significant bit is the ACQHD0R (b0). Once the last byte is written, in the case of Hamming 24/18 check, the result is immediately available on 18 bits, plus one bit for validating the data. In the case of Hamming 8/4 check, the result is available on 4 bits, plus one bit for validating data.

In order to detect uncorrected remaining errors, the ERR bit (bit 23 for Hamming 24/18 or bit 4 for Hamming 8/4) of the bytes word acts as an "Invalid Hamming Data" Flag upon being read (Hamming error detected when the flag is set).

7.7.5.1 Hamming 24/18 Decoding

byte	2								1							
bit	b_{23}	b_{22}	b_{21}	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8
name	P6	D7	D6	D5	D4	D3	D2	D1	P5	M5	M4	M3	M2	M1	A6	A5
byte	0															
bit	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0								
name	P4	A4	A3	A2	P3	A1	P2	P1								

A1-A6 Row or column address
M1-M5 Mode description bits
D1-D6 Data bits
P1-P6 Hamming protection bits

Equation Number	Equation ($\oplus$ = xor)
P1	=b0 $\oplus$ b2 $\oplus$ b4 $\oplus$ b6 $\oplus$ b8 $\oplus$ b10 $\oplus$ b12 $\oplus$ b14 $\oplus$ b16 $\oplus$ b18 $\oplus$ b20 $\oplus$ b22
P2	=b1 $\oplus$ b2 $\oplus$ b5 $\oplus$ b6 $\oplus$ b9 $\oplus$ b10 $\oplus$ b13 $\oplus$ b14 $\oplus$ b17 $\oplus$ b18 $\oplus$ b21 $\oplus$ b22
P3	=b3 $\oplus$ b4 $\oplus$ b5 $\oplus$ b6 $\oplus$ b11 $\oplus$ b12 $\oplus$ b13 $\oplus$ b14 $\oplus$ b19 $\oplus$ b20 $\oplus$ b21 $\oplus$ b22
P4	=b7 $\oplus$ b8 $\oplus$ b9 $\oplus$ b10 $\oplus$ b11 $\oplus$ b12 $\oplus$ b13 $\oplus$ b14
P5	=b15 $\oplus$ b16 $\oplus$ b17 $\oplus$ b18 $\oplus$ b19 $\oplus$ b20 $\oplus$ b21 $\oplus$ b22
P6	=b0 $\oplus$ b1 $\oplus$ b2 $\oplus$ b3 $\oplus$ b4 $\oplus$ b5 $\oplus$ b6 $\oplus$ b7 $\oplus$ b8 $\oplus$ b9 $\oplus$ b10 $\oplus$ b11 $\oplus$ b12 $\oplus$ b13 $\oplus$ b14 $\oplus$ b15 $\oplus$ b16 $\oplus$ b17 $\oplus$ b18 $\oplus$ b19 $\oplus$ b20 $\oplus$ b21 $\oplus$ b22 $\oplus$ b23

Error Bit Address Calculation

The error bit position pos is calculated as follows:

pos = !P5*24+!P4*23+!P3*22+!P2*21+!P1*20

!Pi is Pi complemented.

If pos > 23 and not equal to 0, then there is more than one error in the 3-byte group and the data group is to be rejected. To correct a single-bit error the specified bit is complemented. It is not necessary to correct a protection bit that is in error.

P1-P5	P6	Conclusion	Handling	ERR
all 1	0	No errors in data bits, only in b_{23}	accept data bits	0
all 1	1	No error in data bits	accept data bits	0
don't care 0 & 1	0	single bit error	error bit address calculation see below	0 if *pos* is from 1 to 23, 1 if *pos*= 0 or >23
don't care 0 & 1	1	double bit error	rejected data bits	1

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.5.2 Hamming 8/4 Decoding

Equation Number	Equation ($\oplus$ = xor)
A	= b7 $\oplus$ b5 $\oplus$ b1 $\oplus$ b0
B	= b7 $\oplus$ b3 $\oplus$ b2 $\oplus$ b1
C	= b5 $\oplus$ b4 $\oplus$ b3 $\oplus$ b1
D	= b7 $\oplus$ b6 $\oplus$ b5 $\oplus$ b4 $\oplus$ b3 $\oplus$ b2 $\oplus$ b1 $\oplus$ b0

A	B	C	D	Interpretation	Handling	2^3	2^2	2^1	2^0	ERR bit
0	0	0	0	b_1 erroneous	corrected	b_7	b_5	b_3	!b_1	0
0	0	1	0	b_7 erroneous	corrected	**!** b_7	b_5	b_3	b_1	0
0	1	0	0	b_5 erroneous	corrected	b_7	**!** b_5	b_3	b_1	0
0	1	1	0	b_0 erroneous	accepted	b_7	b_5	b_3	b_1	0
1	0	0	0	b_3 erroneous	corrected	b_7	b_5	**!** b_3	b_1	0
1	0	1	0	b_2 erroneous	accepted	b_7	b_5	b_3	b_1	0
1	1	0	0	b_4 erroneous	accepted	b_7	b_5	b_3	b_1	0
1	1	1	0	b_6 erroneous	accepted	b_7	b_5	b_3	b_1	0
1	1	1	1	no error	accepted	b_7	b_5	b_3	b_1	0
other cases			1	more than 1 error	rejected					1

!bi is bi complemented.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.6 Teletext Signal Quality Measure

In order to determine the quality level of the received CVBS Teletext signal, the following information is provided to the CPU through the Teletext signal Quality Measure register, called "ACQTQM".

7.7.6.1 Bad TV line criterion

During each VBI, "bad" TV lines are counted. The "bad" TV line pulses issued by the Sync Extractor macrocell are counted by a 4 bit counter. These pulses are generated only during the VBI. This "bad TV line" counter is reset by hardware at each Vsync (extracted from the CVBS). If an overflow occurs, then the counter value is locked to Fh.

7.7.6.2 Hamming Errors Code criterion

At the same time as each VBI, the number of Hamming Errors detected by the "8/4 Hamming Decoding" is counted. The detection is done only on the two first bytes after the Framing Code Byte in each Teletext packet during the whole VBI and when the Framing Code is correct and recognized.

One "Hamming Error" pulse corresponds to one or several errors in the same byte, corrected or not by the "8/4 Hamming Decoding". The Hamming Errors are counted by a 4-bit counter, reset by hardware at each Vsync (issued from the CVBS). If an overflow occurs, then the counter value is locked to Fh.

The ACQTQM register provides in a nibble format, the 2 above described kinds of information. The lower nibble of the ACQTQM register gives the number of "bad TV lines" per VBI. The higher nibble of the ACQTQM register gives the number of "Hamming error codes" per VBI.

It is up to the application programmer to use this information and issue an acceptance criterion on the Teletext signal quality to enable or disable any Data Slicing. The ACQTQM register is a read-only register.

7.7.6.3 Reduced Framing Code Criterion

The FR7B status bit in the ACQPRR register provides rough information about the Framing code correctness.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

7.7.7 Register Description

ACQUISITION PAGE OPEN REGISTER (ACQPOR)
R248 - Read/Write
Register Page: 34
Reset value: 0000 0000 (00h)

7							0
POR7	POR6	POR5	POR4	POR3	POR2	POR1	POR0

Bits 7:0 = **POR[7:0]** The data packets associated with each bit will be stored or not depending also on the value if the corresponding MLR bit in the ACQMLR register. Refer to the ACQMLR register description.

ACQUISITION MAGAZINE LOCKED REGISTER (ACQMLR)
R249 - Read/Write
Register Page: 34
Reset value: 0000 0000 (00h)

7							0
MLR7	MLR6	MLR5	MLR4	MLR3	MLR2	MLR1	MLR0

Bits 7:0 = **MLR[7:0]** The associated Data Packets of the corresponding magazine will be stored as described in the following table:

MLRi	PORi	Meaning
1	0	Magazine locked. No storage even if a header is received
0	0	Page closed, wait for new header
0	1	Page opened. Storage

ACQUISITION NEW HEADER RECEIVED REGISTER (ACQNHRR)
R250 - Read/Write
Register Page: 34
Reset value: 0000 0000 (00h)

7							0
NHR7	NHR6	NHR5	NHR4	NHR3	NHR2	NHR1	NHR0

Each bit in the ACQNHR register corresponds to a magazine. It is set by hardware if a new header of the corresponding magazine is received and must be reset by software after each VBI.

Bits 7:0 = **NHR[7:0]** *New Header Received.*
0: No new header received.
1: A new header was received and the storage of the 41 bytes flow depends on the ACQPOR / ACQMLR values shown in the following table.

MLRi	PORi	NHRi	Results
1	0	-	Magazine locked; no storage done
0	0	0	No storage for that magazine during the VBI
0	0	1	Storage started on new header received
0	1	0	Current page not yet closed; data received or not
0	1	1	Current page closed; new page started

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

ACQUISITION PACKET REQUEST REGISTER (ACQPRR)
R251 - Read/Write
Register Page: 34
Reset value: 0xxx 0000 (00h)

7							0
HAMF	-	FR7B	FRDET	SPF	P31	P30	P29

Bit 7 = **HAMF**: *Hamming Check mode control.*
This bit is set and cleared by software.
0: Hamming 8/4 correction
1: Hamming 24/18 correction

Bit 6 = Reserved bit.

Bit 5 = **FR7B**: *Framing Code 7 Matching Bits Error.*
This bit is set by hardware when only 7 of the 8 framing code bits are valid. This bit must be reset by software.
0: No error
1: 7-bits of 8-bit framing code matched

Bit 4 = **FRDET**: *Acknowledge Framing Code 7 Bits.*
This bit is set and cleared by software.
0: Framing codes that match on only 7 bits are not acknowledged.
1: Acknowledge Framing codes that match on only 7 bits

Bit 3 = **SPF**: *Service Packet Flag.*
This bit is set by hardware and cleared by software. The type of service packet requested is programmed using the P[29:31] bits.
0: No service packet received
1: Service packet is received and was requested.

Bits 2:0 = **P[31:P29]**: *X/29, X/30, X/31 Packet Control.*
These bits are programmed by software before each VBI.
0: Service packets Px are never stored.
1: Service packets Px will be stored.

TELETEXT QUALITY MEASURE REGISTER (ACQTQMR)
R252 - Read/Write
Register Page: 34
Reset value: 0000 0000 (00h)

7							0
HEC3	HEC2	HEC1	HEC0	BLC3	BLC2	BLC1	BLC0

Bits 7:4 = **HEC[3:0]**: *Hamming Errors Counter.*
These bits contain the error counter value. When an overflow occurs, the incrementation is stopped at the value Fh. The counter is reset by Hardware at each Vsync. HEC[3] is the most significant bit.

Bits 3:0 = **BLC[3:0]**: *Bad Line Counter.*
These bits contain the error counter value. When an overflow occurs, the incrementation is stopped to the value Fh. The counter is reset by Hardware at each Vsync. BLC[3] is the most significant bit.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

ACQUISITION HAMMING DECODING REGISTER 2 (ACQHD2R)
R253 - Read/Write
Register Page: 34
Reset value: xxxx xxxx (xxh)

7							0
HDE23	HDE22	HDE21	HDE20	HDE19	HDE18	HDE17	HDE16

ACQUISITION HAMMING DECODING REGISTER 1 (ACQHD1R)
R254 - Read/Write
Register Page: 34
Reset value: xxxx xxxx (xxh)

7							0
HDE15	HDE14	HDE13	HDE12	HDE11	HDE10	HDE9	HDE8

ACQUISITION HAMMING DECODING REGISTER 0 (ACQHD0R)
R255- Read/Write
Register Page: 34
Reset value: xxxx xxxx (xxh)

7							0
HDE7	HDE6	HDE5	HDE4	HDE3	HDE2	HDE1	HDE0

These three registers are used for Hamming 24/18 or 8/4 Hamming correction. Refer to the description on the following two pages.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

Hamming 24/18 Correction

On writing, these 24 bits are the data to be Hamming 24/18 checked, where "P" are Hamming bits and "D" are Data bits.

ACQHD2R

23							16
P6	D18	D17	D16	D15	D14	D13	D12

ACQHD1R

15							8
P5	D11	D10	D9	D8	D7	D6	D5

ACQHD0R

7							0
P4	D4	D3	D2	P3	D1	P2	P1

On reading, the bits are redefined as follows:

ACQHD2R

23							16
ERR	D18	D17	D16	D15	D14	D13	D12

ACQHD1R

15							8
0	0	0	D11	D10	D9	D8	D7

ACQHD0R

7							0
0	0	D6	D5	D4	D3	D2	D1

Bit 23 = **ERR:** *Hamming Error.*
This bit is set by hardware and cleared by software.
0: The 24-bit word is correctly Hamming decoded (no error or a recoverable error).
1: The result of the Hamming decoding is wrong (unrecoverable error).

Note: The D[18:1] bits are cleared by hardware when ERR=1.

Bits 22:16= **D[18:12]:** *Data*

Bits 15:13 = Reserved

Bits 12:8 = **D[11:7]:** *Mode*

Bits 7:6 = Reserved

Bits 5:0 = **D[6:1]:** *Address*

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

Hamming 8/4 correction:

HDE[7:0]: on writing, these 8 bits are the data to be Hamming 8/4 checked, where P are Hamming bits D are Data bits.

HDE[23:8]: don't care.

On reading, the bits are redefined as follows:

ACQHD2R

7							0
X	X	X	X	X	X	X	X

ACQHD2R

23							16
0	0	0	0	0	0	0	0

ACQHD1R

7							0
X	X	X	X	X	X	X	X

ACQHD1R

15							8
0	0	0	0	0	0	0	0

ACQHD0R

7							0
D4	P	D3	P	D2	P	D1	P

ACQHD0R

7							0
0	0	0	ERR	D4	D3	D2	D1

Bits 23:5 = Don't care

Bit 4 = **ERR:** *Hamming Error.*
This bit is set by hardware and cleared by software.
0: The 24-bit word is correctly Hamming decoded (no error or a recoverable error).
1: The result of the Hamming decoding is wrong (unrecoverable error).

Note: The D[4:1] bits are cleared by hardware when ERR=1.

Bits 3:0 = D4-D1: Data

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

The ACQAD1R, ACQAD0R registers are 2 read/ write registers that define the current address value, in 13 bits as follows:

ACQUISITION ADDRESS REGISTER 1 (ACQAD1R)
R242 - Read/Write
Register Page: 34
Reset value : 000x xxxx (xxh)

7							0
0	0	0	AD12	AD11	AD10	AD9	AD8

Bits 7:5 = Reserved bits. They must be left at their reset value.

Bits 4:0 = **AD[12:8]**: The register content is the address of the next free location in TDSRAM. The address is automatically incremented by hardware. It could be modified by software after each VBI.

ACQUISITION ADDRESS REGISTER 0 (ACQAD0R)
R243 - Read/Write
Register Page: 34
Reset value: xxxx xxxx (xxh)

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Bits 7:0 = **AD[7:0]**: The register content is the address of the next free location in TDSRAM. The address is automatically incremented by hardware. It could be modified by software after each VBI.

TELETEXT SLICER CONTROL REGISTER (TXSCR)
R248 - Read/Write
Register Page: 35
Reset value: 0000 0110 (06h)

7							0
TXTEN	TXTES	TXTGC 1	TXTGC 0	-	1	1	0

Bit 7 = **TXTEN:** *Teletext slicer enable.*
0: Disables the Teletext Slicer; all analog functions are in zero power consumption (reset state).
1: Starts the Slicer function.

Bit 6 = **TXTES:** *Teletext extended search.*
0: Run-in Clock search starts from line 6 or 318.
1: Run-in Clock search starts at line 2 (first field) and line 315 (second field).

Bits 5:4 = **TXTGC:** Teletext Pre-amplifier Gain Control bits. These two bits control the effective gain of the pre-amplifier stage inserted between the CVBS1 signal input and the Teletext Slicer comparator input. (Refer to the ADSEL bit in the PASR register)

TXTGC1	TXTGC0	Pre-amplifier Gain Value
0	0	1.2
0	1	2.0
1	0	3.0
1	1	3.8

Bit 3 = Reserved (read only).

Bits 2:0 = Reserved bits, keep in reset state.

TELETEXT SLICER AND ACQUISITION UNIT (Cont'd)

TELETEXT SLICER INITIALIZATION REGISTER (TXSLIR)
R249 - Read/Write
Register Page: 35
Reset Value: 0000 1011 (0Bh)

7							0
0	0	INI5	INI4	INI3	INI2	INI1	INI0

Bits 7:6 = Reserved bits. Keep in reset state.

Bit 5:0 = **INI(5:0**): 6-bit initialization value of the Slicer DAC. This value is used to re-initialize the Slicer DAC at each beginning of TV line.

PRE-AMPLIFIER AND ADC SELECT REGISTER (PASR)
R255 **-** Read/Write
Register Page: 35
Reset Value: 0000 0000 (00h)

7							0
0-	0	0	0	0	PRE-AMP	0	ADC-SEL

Bits 7:3 = Reserved bits, keep in reset state.

Bit 2 = **PREAMP:** *Preamplifier enable.*
This bit is set and cleared by software to define the analog CVBS input configuration of the Teletext slicer.
0: Activate on-chip pre-amplifier providing to the Teletext front-end Slicer an amplified CVBS signal, the gain being defined by the TXTGC(1:0) bits of the Teletext Slicer Control Register (TXSCR).
1: Pre-amplifier is by-passed.

Bit 1 = Reserved bit, keep in reset state.

Bit 0 = **ADCSEL:** *ADC Selection.*
This bit is set and cleared by software to choose which analog signal is sent to the A/D converter channel 0.
0: Channel 0 is connected to the VDETB signal (amplitude of the sync pulses extracted from the CVBS signal).
1: Channel 0 is connected to the VBGP signal (on-chip analog reference voltage).

The Sync pulse amplitude seen at the CVBS1 pin can be easily determined calculating the difference of 2 A/D conversions, converting first the VDETB signal and then the VBGP signal (the maximum code FFh represents a 1V amplitude).

VsyncAmp= (N_{VDETB}-N_{VBGP})/FF Volt

For an ideal composite video signal, the theoretical gain to program with the TXTGC(1:0) bits is:
Gth= 0.4 / VsyncAmp (V)

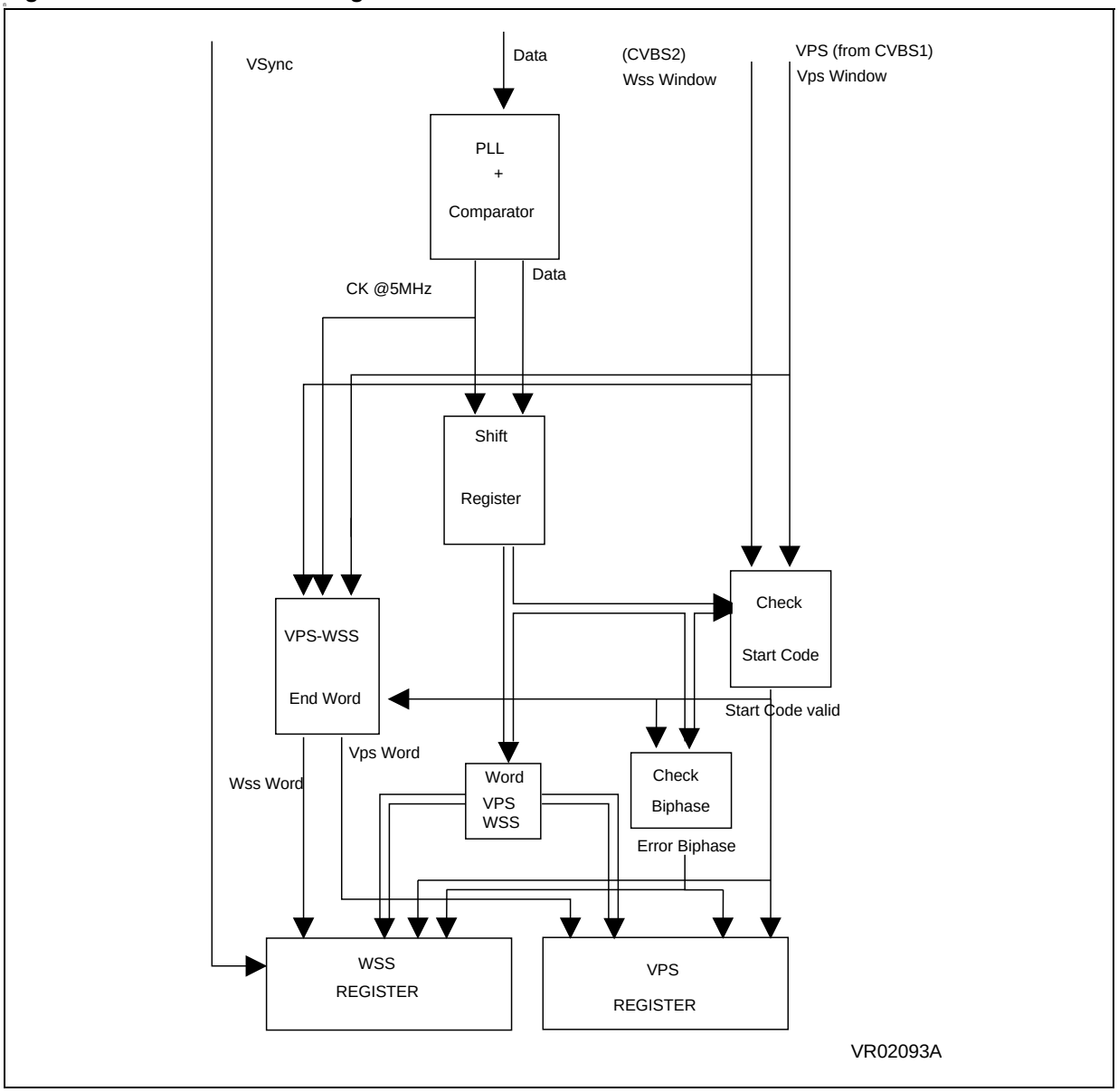
7.8 VPS & WSS SLICER

7.8.1 Introduction

The VPS/WSS digital sections perform several functions:

- Separate the horizontal and vertical sync signals
- Determine the field and line information (in conjunction with the Sync Extractor)
- Process the data stream
- Generate timing and windows for datastream information extraction
- Load data into the registers and report any bi-phase errors found on any bytes (each word of information has an error report to help increase performance, especially during weak signals)

Figure 92. VPS/WSS Block Diagram



VPS & WSS SLICER (Cont'd)

7.8.2 General Operation

The VPS/WSS Slicer implements the following features:

Wide Screen Signalling (WSS) follows the proposed ETSI standard providing advanced television features such as aspect ratio and format using 14 Bits of information extracted from Half Line 23, Field 1.

Video Programming System (VPS) follows the industry standard for VPS using 5 bytes of information extracted from Line 16, on Field 1 (bytes 5, 11 to 14).

The input signals are treated internally by the system as follows:

– **Data:** the serial data from the analog video input is converted to serial digital data for digital processing.

– **WSS window/VPS window.** For the VPS signal treatment, the analog video signal is taken from the CVBS1 pin after its amplification through the programmable amplifier. For the WSS signal treatment, the video signal is taken from the CVBS2 pin directly. The CVBS2 pin is AC driven through a 150pF capacitor.

– **VSync:** The system also receives the sync signals from the video input pin. This signal contains the information telling exactly which line and field is being processed. This is performed in the Sync Extractor Cell with its outputs delivered to the VPS/WSS cell.

7.8.2.1 Analog Input stage

The data slicer determines the average amplitude of the input data stream and references a comparator at this value. The average amplitude is accomplished by AC coupling the video input signal into the slicer cell and gating the bias circuits during the active bi-phase signal.

7.8.2.2 Video Input Specification

The composite video baseband signal (CVBS) input is 1 V_{PP} +/- 6dB from the bottom of sync to 100IRE video for both CVBS1 (VPS path) and CVBS2 (WSS path). The polarity of the video is negative going sync.

7.8.2.3 Digital Decoder

The digital decoder uses the 5MHz clock as the main timing reference and the digital outputs from the sync and data slicer as the signal inputs. The data for both VPS and WSS are transmitted with a reference of 5 MHz clock with the same phase.

The serial digital data coming out of the data slicer is first sampled at 5 MHz, this oversamples the 2.5Mbits/sec enough to process the signal under normal conditions. The samples are then digitally processed.

The bi-phase code is checked on both sides of the signal with respect to a timing delay to detect errors.

7.8.2.4 Framing Code and Data Check

If the framing code is recognized, the data is further processed and a flag is set.

The data is then processed for biphase errors. If the data is not defective, bits ERB1..5 remain at 0 and the data is stored. If the data is in error, bits ERB1..5 are set by hardware. These bits must be cleared by the user before the next framing code.

If a framing code is not recognized, the data is not stored , the flag remains at 0 (VFRMVPS bit in the VPSSR register) and the slicer waits for the next VPS/WSS window.

VPS & WSS SLICER (Cont'd)

7.8.3 About Video Programming System (VPS)

The VPS extracts the digital data from line 16 of a transmitted PAL video signal. The VPS feature for the VCR uses only 5 of the 13 bytes of information to help simplify the programming of the VCR for recording.

A total of 15 words is transmitted on field 1, line 16. The signal is bi-phase modulated at a 2.5Mbit/s rate. The VPS transmission specification defines the clock frequency is 5MHz (+/-.01%), the signal shape is sine squared pulses with half amplitude pulse duration equal to 200ns +/- 10ns, and the signal amplitude is 0.5V +/- 5%.

The first word starts 12.5 +/- 1.5 sync pulse. The words (bytes) of data are recalled below:

Word 1: Run-in (10.10.10.10.10.10.10.10)

The run-in word is a group of all 1's. With the bi-phase coding the signal is 1010101010101010 that is used for clock synchronisation inside the VPS cell.

Word 2: Start code (10.00.10.10.10.01.10.01)

The start code helps to define the presence of VPS and to synchronize the following data stream. If the start code has not been recognised 12.7 +/- 0.2 the VPSWin rising edge, the system waits for the next VPSWin to find a Start code.

Words 3 and 4 are not relevant for the VPS coding.

Word 5: Identification of sound channels and "Rated R" mark

VPS Bit #	-	-	-	-	-	-	-	-
Word 5	7	6	5	4	3	2	1	0
	Sound	Sound	R-rating	R-rating	reserve	reserve	reserve	reserve
VPS01	SD(1)	SD(0)	RR(1)	RR(0)	RX(3)	RX(2)	RX(1)	RX(0)
	Sound Bits		R-Rating Bits		Reserved Bits for future application			

Words 6 through 10 are not relevant for VPS coding.

Words 11 through 14 contain 32 bits. Complete information may involve bits contained in two bytes. These bits are numbered from 0 to 31 in the VPS transmission spec to show the order in reverse time that the bits are actually transmitted. This means the MSB is the first bit following the framing code for any one grouping of information. The bit numbers are included below to help cross-reference the bits with the VPS transmission specification.

Word 15 is not relevant for VPS coding and is not defined in the VPS specification.

VPS Bit #	0	1	2	3	4	5	6	7
Word 11	7	6	5	4	3	2	1	0
	ID	ID	Day	Day	Day	Day	Day	Month
VPS02	ID(1)	ID(0)	ADT(4)	ADT(3)	ADT(2)	ADT(1)	ADT(0)	AMT(3)

ID of address Announced Day of Transmission (6 bits) (binary)

VPS Bit #	8	9	10	11	12	13	14	15
Word 12	7	6	5	4	3	2	1	0
	Month	Month	Month	Hour	Hour	Hour	Hour	Hour
VPS03	AMT(2)	AMT(1)	AMT(0)	ASH(4)	ASH(3)	ASH(2)	ASH(1)	ASH (0)

Announced Month of Transmission (4 bits) Announced Start Hour (5 bits) binary

VPS Bit #	16	17	18	19	20	21	22	23
Word 13	7	6	5	4	3	2	1	0
	Minute	Minute	Minute	Minute	Minute	Minute	Nation	Nation
VPS04	ASM(5)	ASM(4)	ASM(3)	ASM(2)	ASM(1)	ASM(0)	NC (3)	NC (2)

Announced Start Minute (6 bits) Nationality (4 bits) (binary)

VPS Bit #	24	25	26	27	28	29	30	31
Word 14	7	6	5	4	3	2	1	0
	Nation	Nation	Source	Source	Source	Source	Source	Source
VPS05	NC (1)	NC (0)	PSC(5)	PSC(4)	PSC(3)	PSC(2)	PSC(1)	PSC(0)

Nationality (4 bits) Program Source Code (6 bits)

VPS & WSS SLICER (Cont'd)

7.8.4 About Wide-Screen-Signaling (WSS)

The main purpose of the WSS signal is to define the aspect ratio and format of the current television signal. This signal is needed with the use of 16:9 displays where both 4:3 and 16:9 transmissions may be present at the same time on different channels. The information contained in WSS defines the transmitted aspect ratio and the recommended display format so that the television will automatically adjust the display for optimal viewing.

The WSS signal contains digital data on the first half of line 23 of a transmitted PAL/SECAM video signal. The WSS signal is intended to be recorded on a video cassette recorder (VCR) during normal recording to keep the encoded aspect ratio information together with the program of interest. The automatic switching of the television would occur during the playback of the previously recorded program. In order to support double WSS writing, the system may slice on both lines 22 and 23. The registers have been duplicated to deliver a second line of WSS if available.

The signal is defined by 137 "200ns" elements for a total of 27.4µs and is one third the VPS rate to allow recovery of data from VCR tapes. WSS has an effective clock of only 1.66MHz but is still defined using the same 5MHz clock used in the VPS system.

7.8.5 WSS Signal Content

The parts of the signal are summarized below:

Word 1: Run-in
(1.1111.0001.1100.0111.0001.1100.0111)
29 elements @ 5MHz

The run-in word is 1F1C71C7(h) using a 5MHz clock.

Word 2: Start code
(0001.1110.0011.1100.0001.1111)
24 elements @ 5MHz

The start code helps to define the presence of WSS to synchronize the following data stream. The signal consists of 1E3C1F(h) or 000111100011110000011111 using a 5MHz clock.

If the start code is not recognised 14.3 +/- 0.2 µs after the WSSWin rising edge, the system waits for the next WSSWin to find a Start code.

Data Bits Purpose Bi-phase Encoding Duration(1.2

Data Group 1 Bits:

B(0) Aspect Ratio Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(1) Aspect Ratio Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(2) Aspect Ratio Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(3) Aspect Ratio Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

Data Group 2 Bits:

B(4) Enhanced Services Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(5) Enhanced Services Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(6) Enhanced Services Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(7) Enhanced Services Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

Data Group 3 Bits:

B(8) Subtitles Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(9) Subtitles Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(10) Subtitles Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

Data Group 4 Bits:

B(11) Reserved Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(12) Reserved Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

B(13) Reserved Bit
(0=000111 or 1=111000) 6 elements @ 5MHz

The WSS signal ends on the first half of line 23. Total time = 137 elements @ 5 MHz. The second half of line 23 contains video that is not related to WSS.

VPS & WSS SLICER (Cont'd)

7.8.6 WSS Data Group Assignments

The 14 data bits of the WSS signal are divided into 4 groups. Group 1 contains 4 bits in which the first 3 carry data and the last bit is the odd parity bit for the first three bits. Group 2 contains 4 data bits. Group 3 and Group 4 contain 3 data bits.

Group 1 Data: Aspect Ratio Bits

Bits 0, 1, and 2 are data bits with Bit 3 contains the odd parity bit. The data bits in Group 1 are defined in Table 30 Group1 Data Bits.

Group 2 Data: Enhanced Service Bits

B(4) Film bit defined by: '0' is camera mode, '1' is film mode.

B(7)-B(5)

Reserved. Set to "0".

Group 3 Data: Subtitle Bits

B(8) This bit denotes if subtitles are within the teletext.

When "0"; Subtitles are NOT found within teletext.

When "1"; Subtitles are found within teletext.

B(9) and B(10) are Subtitling mode bits defined in the following table:

b10	b9	b8	Location of Subtitles
x	x	0	No Subtitles in teletext
0	0	1	No open subtitles
0	1	1	Subtitles are in the active image area
1	0	1	Subtitles are out of the active image area
1	1	1	Reserved

Group 4 Data: Reserved Bits

B(13)-B(11)

These bits are reserved. They are set to "0".

Table 30. Group1 Data Bits

B3	B2	B1	B0	Aspect Ratio	Aspect Ratio Range	Format	Position	Number of Active Lines (estimated)
1	0	0	0	4:3	a<= 1.46	Full	Not applicable	576
0	0	0	1	14:9	1.46 < a <= 1.66	Letterbox	center	504
0	0	1	0	14:9	1.46 < a <= 1.66	Letterbox	top	504
1	0	1	1	16:9	1.66 < a <= 1.90	Letterbox	center	430
0	1	0	0	16:9	1.66 < a <= 1.90	Letterbox	top	430
1	1	0	1	>16:9	1.90 < a	Letterbox	center	not defined
1	1	1	0	14:9	1.46 < a <= 1.66	Full (*)	center	576
0	1	1	1	16:9	1.66 < a <= 1.90	Full	Not applicable	576

The number of active lines is only an indication for the exact aspect ratios a=1.33, a=1.56, and a=1.78.

(*) The actual transmitted aspect ratio is 4:3 but a 14:9 window should contain all the relevant picture content to encourage a wide screen display on a 16:9 television set.

VPS & WSS SLICER (Cont'd)

7.8.7 Register Description

VPS STATUS REGISTER (VPSSR)
R240 - Read Only/Reset by writing
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
-	NWDATVP	VFRMVPS	ERB5	ERB4	ERB3	ERB2	ERB1

Bit 7 = Reserved.

Bit 6 = **NWDATVP:** *New data* in all five data registers. Data will be written into this register with or without bi-phase errors. The error bit for each byte of data should be checked to validate any data. Reset by writing to this register or by the leading edge of VPS Window. This flag will only be set when the number of bits saved is correct for the time expected for those bits.

Bit 5 = **VFRMVPS:** *Valid Framing code for VPS*. This bit is set when the framing code is error free. Reset by the leading edge of VPS window.

Bit 4 = **ERB5:** *Error in Byte for VPSD4R*
Data was written but a bi-phase error was detected. Set by hardware when there is an error. The user must reset it by software.

Bit 3 = **ERB4**: *Error in Byte for VPSD3R*
Data was written but a bi-phase error was detected. Set by hardware when there is an error. The user must reset it by software.

Bit 2= **ERB3:** *Error in Byte for VPSD2R*
Data was written but a bi-phase error was detected. Set by hardware when there is an error. The user must reset it by software.

Bit 1 = **ERB2:** *Error in Byte for VPSD1R* (R/W)
Data was written but a bi-phase error was detected. Set by hardware when there is an error. The user must reset it by software.

Bit 0 = **ERB1:** *Error in Byte for VPSD0R* (R/W)
Data was written but a bi-phase error was detected. Set by hardware when there is an error. The user must reset it by software

VPS DATA REGISTER 0 (VPSD0R)
R241 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
SD1	SD0	RR1	RR0	RX3	RX2	RX1	RX0

Bit 7:6 = **SD[1:0]**: *Sound Bits*.

Bit 5:4 = **RR[1:0]:** *R-Rating bits* for flag to show the material may not be suitable for children.

Bit 3:0 = **RX[3:0]**: Reserved bits for future sound applications. Presently not defined.

VPS DATA REGISTER 1 (VPSD1R)
R242 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
ID1	ID0	ADT4	ADT3	ADT2	ADT1	ADT0	AMT3

Bit 7:6 = **ID[1:0]:** *Identification of the address* for this VPS line.

Bit 5:1 = **ADT[4:0]:** *Announced day of transmission*.

Bit 0 = **AMT3:** *Announced month of transmission*.

VPS DATA REGISTER 2 (VPSD2R)
R243 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
AMT2	AMT1	AMT0	ASH4	ASH3	ASH2	ASH1	ASH0

b7:5 = **AMT[2:0]:** *Announced month of transmission*.

Bit 4:0 = **ASH[4:0]:** *Announced start of item, HOUR*.

VPS & WSS SLICER (Cont'd)

VPS DATA REGISTER 3 (VPSD3R)
R244 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
ASM5	ASM4	ASM3	ASM2	ASM1	ASM0	NC3	NC2

Bit 7:2 = **ASM[5:0]**: *Announced start of item, MINUTE*.

Bit 1:0 = **NC[3:2]**: Nationality code which is used to identify the source of the item.

VPS DATA REGISTER 4 (VPSD4R)
R245 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
NC1	NC0	PSC5	PSC4	PSC3	PSC2	PSC1	PSC0

Bit 7:6 = **NC[1:0]**: Nationality code which is used to identify the source of the item.

Bit 5:0 = **PS[5:0]**: *Program source code* used to identify the source of the item inside the country identified in the nationality code.

WSS DATA AND STATUS REGISTER 0 (WSSDS0R)
R246 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
x	NWDATWS	VFRMWS	GP1ERF	WSS3	WSS2	WSS1	WSS0

Bit 7 = Reserved.

Bit 6 = **NWDATWS**: *New data* in all three data registers. Data will be written into this register with or without bi-phase errors. The error bit for each byte of data should be checked to validate any data. Reset by writing to this register or by the leading edge of Wss Window. This flag will only be set when the number of bits saved is correct for the time expected for those bits.

Bit 5= **VFRMWSS:** *Valid Framing code for WSS*. This bit is set when the framing code is error free. Reset by the end of WSS Window.

Bit 4 = **GP1ERF**: This bit is set when any of the Group 1 bits (WSS3-0) are received with a bi-phase error.

b3-b0 = **WSS3-0:** *WSS Aspect Ratio Bits*

WSS DATA AND STATUS REGISTER 1 (WSSDS1R)
R247 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
WSS2X	x	x	GP2ERF	WSS3	WSS2	WSS1	WSS0

Bit 7 = **WSS2X:** WSS has been received 2X (2 times). This bit is set when a second line of WSS has been saved in registers VPS0A - VPS0C.

Bit 6:5 = Reserved.

Bit 4 = **GP2ERF:** This bit is set when any of the Group 2 bits (WSS[3:0]) is received with a bi-phase error.

Bit 3:0 = **WSS[7:4]**: WSS Enhanced Service Bits.

WSS DATA AND STATUS REGISTER 2 (WSSDS2R)
R248 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
GP3ERF	WSS10	WSS9	WSS8	GP4ERF	x	x	x

Bit 7 = **GP3ERF:** This bit is set when any of the Group 3 bits (WSS[3-0]) are received with a bi-phase error.

Bit 6:4 = **WSS[10:8]:** WSS Subtitle Bits

Bit 3 = **GP4ERF:** This bit is set when any of the Group 4 bits (WSS[3:0]) are received with a bi-phase error.

Bit 2:0 = Reserved Bits

VPS & WSS SLICER (Cont'd)

VPS/WSS CONTROL REGISTER (VPSWSSCR)
R249 - Read/Write
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
EN-CELL	VCRWIN	BIAS-WIN	VW_EN	WIND-LY	0	0	0

Bit 7 = **ENCELL:** *VPS/WSS enable*.
0 : WSS/VPS disabled (reset condition)
1 : WSS/VPS enabled

Bit 6 = **VCRWIN:** When this bit is set, the line select window for VPS becomes three lines wide (15-17) while the line select for WSS becomes four lines wide (21-24). The four lines are required due to the uncertainty of the line number from a VCR source and the repeated WSS signal found on both lines 22 and 23.

When the bit is reset, VPS is active only on line 16 and WSS is active only on line 23.

Bit 5 = **BIASWIN:** *Bias Window* is a control bit used to keep the WSS and VPS bias window open during the entire line. When set, the window is active during the run-in and the data periods. This feature may be used during poor signals or as a backup due to leakage and noise problems on the IC.

When reset, the window is defin

ed only for the run-in clock.

Bit 4 = **VW_EN**: *Clock Generating Unit Enable Bit*
0: No clock is provided to the VPS/WSS slicer unit.
1: The VPS/WSS clock is provided by the Digital Clock Generating Unit based on the 48.56 MHz issued by the TXT frequency multiplier.

Note: This bit is valid only when the SLIEN bit of the SLCCR register is set (refer to the Timing and Clock Controller Chapter), i.e. when the PLL is enabled.

Bit 3 = **WINDLY**: *Window Delay Bit*
This bit is common to VPS and WSS. Due to the large range of signals in VPS mode (data start between 11ms and 14ms) the window which allows to adjust the slicing level during the run-in clock cannot be the same for early and late signals. This bit has to be controlled to get a proper bias window in VPS.
0: Run-in bias window defined for standard & too late data (VPS data start between 12.35ms and 14ms).
1: Run-in bias window defined for too early data (VPS data start between 11ms and 12.65ms)

Note: This bit has no influence on WSS bias window.

Bit 2:0 Reserved bits.

DUPLICATE WSS STATUS AND DATA REGISTER (WSSDS3R)
R250 - Read Only/Reset by writing
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
x	NWDATWS2	VFRMWSS2	GP1ERF2-	WSS2(3)	WSS2(2)	WSS2(1)	WSS2(0)

Bit 7 = Reserved.

Bit 6 = **NWDATWS2:** *New data* in all three data registers. Data will be written into this register with or without bi-phase errors. The error bit for each byte of data should be checked to validate any data. Reset by writing any value to this register or by the leading edge of WSS Window, that initiates a new acquisition.

Bit 5 = **VFRMWSS2:** Valid Framing code for WSS. This bit is set when the framing code is error free. Reset by the end of WSS window.

Bit 4 = **GP1ERF2**: This bit is set when any of the Group 1 bits (WSS3-0) are received with a bi-phase error.

Bit 3:0 = **WSS2[3:0]:** WSS Aspect Ratio Bits

VPS & WSS SLICER (Cont'd)

DUPLICATE WSS DATA AND STATUS REGISTER (WSSDS4R)
R251- Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
WSS2X	x	x	GP2ERF2	WSS2(7)	WSS2(6)	WSS2(5)	WSS2(4)

Bit 7 = **WSS2X**: WSS has been received 2X (2 times). This bit is set when a second line of WSS has been saved in registers VPS0A - VPS0C.

Bit 6:5 = Reserved.

Bit 4 = **GP2ERF2**: This bit is set when any of the Group 2 bits (WSS[3:0]) are received with a bi-phase error.

Bit 3:0 = **WSS2[7:4]**: *WSS Enhanced Service Bits*

DUPLICATE WSS DATA REGISTER (WSSDS5R)
R252 - Read Only
Register Page: 6
Reset value: 0000 0000 (00h)

7							0
GP3 ERF2	WSS2(10)	WSS2(9)	WSS2(8)	GP4 ERF2	x	x	x

Bit 7 = **GP3ERF2**: This bit is set when any of the Group 3 bits WSS[3:0] are received with a bi-phase error.

Bit 6:4 = **WSS2[10:8]**: WSS Subtitle Bits

Bit 3 = **GP4ERF2**: This bit is set when any of the Group 4 bits (WSS[3:0]) are received with a bi-phase error.

Bit 2:0 = Reserved bits.

7.9 SERIAL PERIPHERAL INTERFACE (SPI)

7.9.1 Introduction

The Serial Peripheral Interface (SPI) is a general purpose on-chip shift register peripheral. It allows communication with external peripherals via an SPI protocol bus.

In addition, special operating modes allow reduced software overhead when implementing I^2C-bus and IM-bus communication standards.

The SPI uses up to 3 pins: Serial Data In (SDI), Serial Data Out (SDO) and Synchronous Serial Clock (SCK). Additional I/O pins may act as device selects or IM-bus address identifier signals.

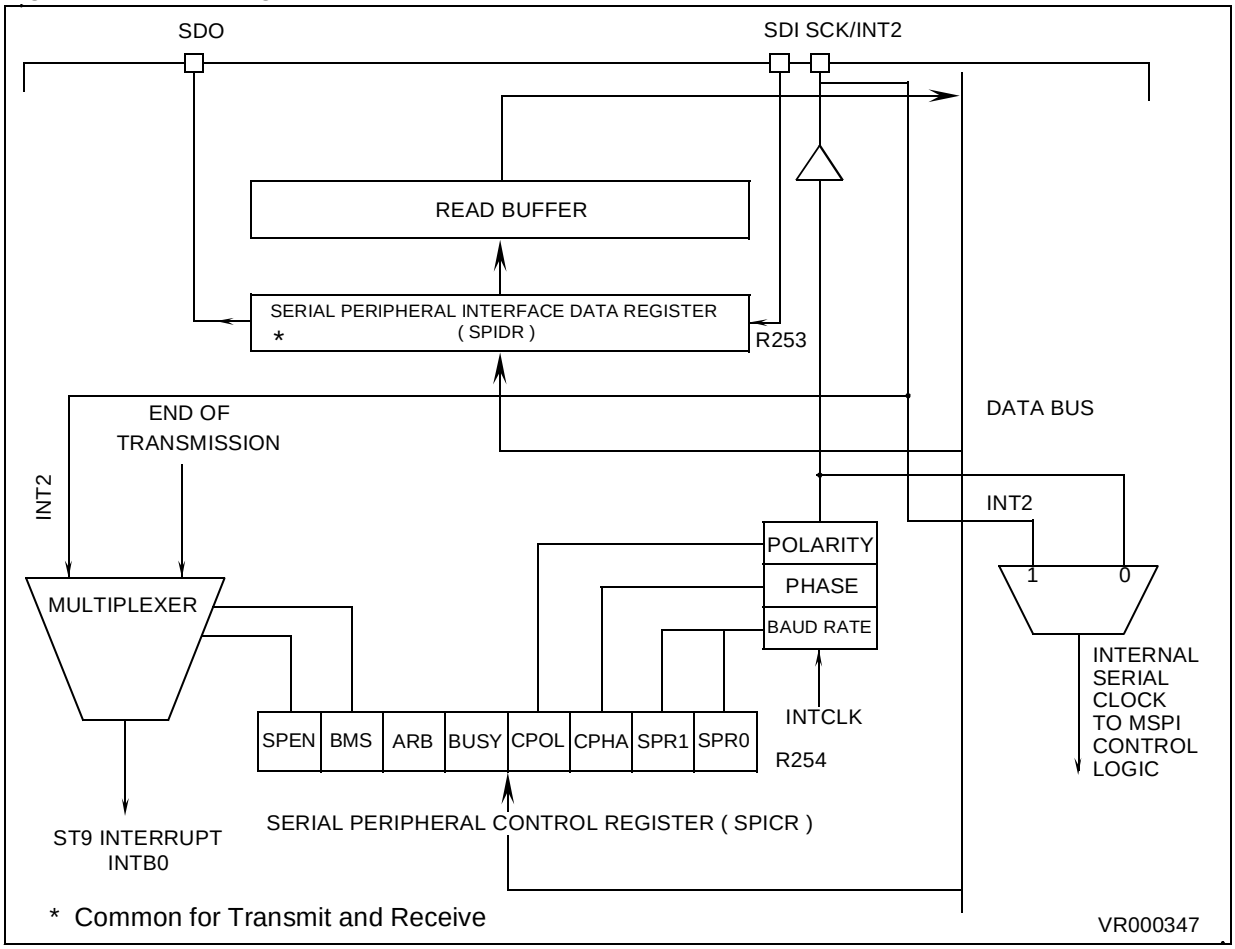
The main features are:

- Full duplex synchronous transfer if 3 I/O pins are used
- Master operation only
- 4 Programmable bit rates
- Programmable clock polarity and phase
- Busy Flag
- End of transmission interrupt
- Additional hardware to facilitate more complex protocols

7.9.2 Device-Specific Options

Depending on the ST9 variant and package type, the SPI interface signals may not be connected to separate external pins. Refer to the Peripheral Configuration Chapter for the device pin-out.

Figure 93. Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.9.3 Functional Description

The SPI, when enabled, receives input data from the internal data bus to the SPI Data Register (SPIDR). A Serial Clock (SCK) is generated by controlling through software two bits in the SPI Control Register (SPICR). The data is parallel loaded into the 8 bit shift register during a write cycle. This is shifted out serially via the SDO pin, MSB first, to the slave device, which responds by sending its data to the master device via the SDI pin. This implies full duplex transmission if 3 I/O pins are used with both the data-out and data-in synchronized with the same clock signal, SCK. Thus the transmitted byte is replaced by the received byte, eliminating the need for separate "Tx empty" and "Rx full" status bits.

When the shift register is loaded, data is parallel transferred to the read buffer and becomes available to the CPU during a subsequent read cycle.

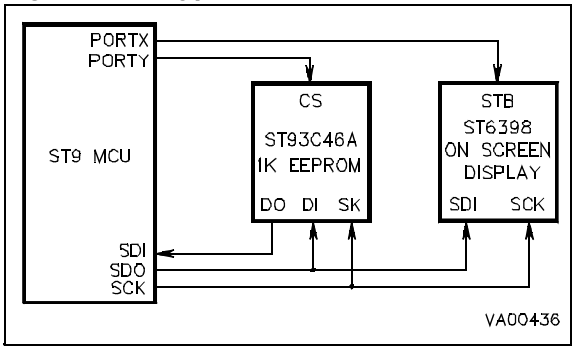
The SPI requires three I/O port pins:

SCK	Serial Clock signal
SDO	Serial Data Out
SDI	Serial Data In

An additional I/O port output bit may be used as a slave chip select signal. Data and Clock pins I²C Bus protocol are open-drain to allow arbitration and multiplexing.

Figure 93 below shows a typical SPI network.

Figure 94. A Typical SPI Network



7.9.3.1 Input Signal Description

Serial Data In (SDI)

Data is transferred serially from a slave to a master on this line, most significant bit first. In an S-BUS/I^2C-bus configuration, the SDI line senses the value forced on the data line (by SDO or by another peripheral connected to the S-bus/I^2C-bus).

7.9.3.2 Output Signal Description

Serial Data Out (SDO)

The SDO pin is configured as an output for the master device. This is obtained by programming the corresponding I/O pin as an output alternate function. Data is transferred serially from a master to a slave on SDO, most significant bit first. The master device always allows data to be applied on the SDO line one half cycle before the clock edge, in order to latch the data for the slave device. The SDO pin is forced to high impedance when the SPI is disabled.

During an S-Bus or I^2C-Bus protocol, when arbitration is lost, SDO is set to one (thus not driving the line, as SDO is configured as an open drain).

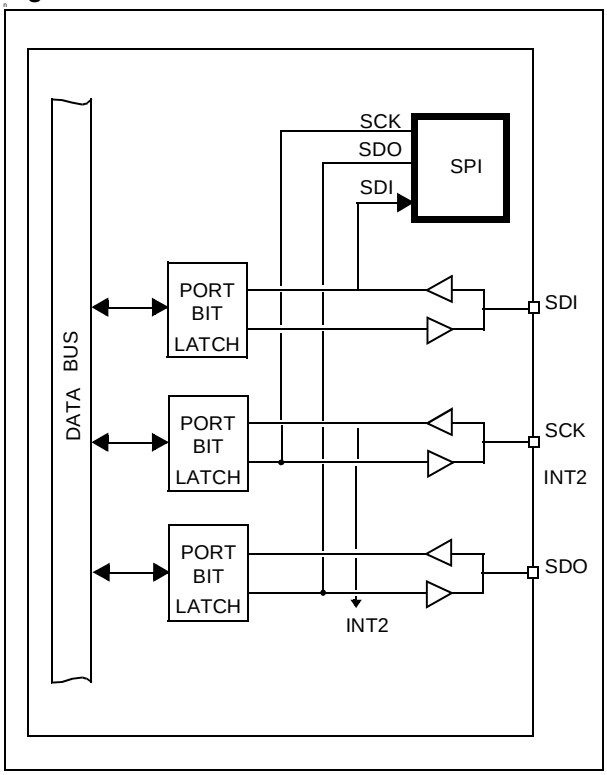
Master Serial Clock (SCK)

The master device uses SCK to latch the incoming data on the SDI line. This pin is forced to a high impedance state when SPI is disabled (SPEN, SPICR.7 = "0"), in order to avoid clock contention from different masters in a multi-master system. The master device generates the SCK clock from INTCLK. The SCK clock is used to synchronize data transfer, both in to and out of the device, through its SDI and SDO pins. The SCK clock type, and its relationship with data is controlled by the CPOL (Clock Polarity) and CPHA (Clock Phase) bits in the Serial Peripheral Control Register (SPICR). This input is provided with a digital filter which eliminates spikes lasting less than one INTCLK period.

Two bits, SPR1 and SPR0, in the Serial Peripheral Control Register (SPICR), select the clock rate. Four frequencies can be selected, two in the high frequency range (mostly used with the SPI protocol) and two in the medium frequency range (mostly used with more complex protocols).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 95. SPI I/O Pins



7.9.4 Interrupt Structure

The SPI peripheral is associated with external interrupt channel B0 (pin INT2). Multiplexing between the external pin and the SPI internal source is controlled by the SPEN and BMS bits, as shown in Table 31 Interrupt Configuration.

The two possible SPI interrupt sources are:

– End of transmission (after each byte).

– S-bus/I^2C-bus start or stop condition.

Care should be taken when toggling the SPEN and/or BMS bits from the "0,0" condition. Before changing the interrupt source from the external pin to the internal function, the B0 interrupt channel should be masked. EIMR.2 (External Interrupt Mask Register, bit 2, IMBO) and EIPR.2 (External Interrupt Pending Register bit 2, IMP0) should be "0" before changing the source. This sequence of events is to avoid the generating and reading of spurious interrupts.

A delay instruction lasting at least 4 clock cycles (e.g. 2 NOPs) should be inserted between the SPEN toggle instruction and the Interrupt Pending bit reset instruction.

The INT2 input Function is always mapped together with the SCK input Function, to allow Start/Stop bit detection when using S-bus/I^2C-bus protocols.

A start condition occurs when SDI goes from "1" to "0" and SCK is "1". The Stop condition occurs when SDI goes from "0" to "1" and SCK is "1". For both Stop and Start conditions, SPEN = "0" and BMS = "1".

Table 31. Interrupt Configuration

SPEN	BMS	Interrupt Source
0	0	External channel INT2
0	1	S-bus/I^2C bus start or stop condition
1	X	End of a byte transmission

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.9.5 Working With Other Protocols

The SPI peripheral offers the following facilities for operation with S-bus/I^{2}C-bus and IM-bus protocols:

- Interrupt request on start/stop detection
- Hardware clock synchronisation
- Arbitration lost flag with an automatic set of data line

Note that the I/O bit associated with the SPI should be returned to a defined state as a normal I/O pin before changing the SPI protocol.

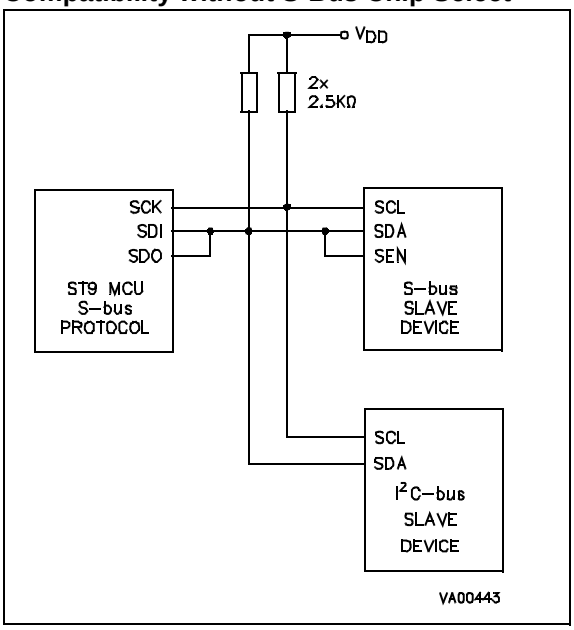
The following paragraphs provide information on how to manage these protocols.

7.9.6 I^{2}C-bus Interface

The I^{2}C-bus is a two-wire bidirectional data-bus, the two lines being SDA (Serial DAta) and SCL (Serial CLock). Both are open drain lines, to allow arbitration. As shown in Figure 96, data is toggled with clock low. An I²C bus start condition is the transition on SDI from 1 to 0 with the SCK held high. In a stop condition, the SCK is also high and the transition on SDI is from 0 to 1. During both of these conditions, if SPEN = 0 and BMS = 1 then an interrupt request is performed.

Each transmission consists of nine clock pulses (SCL line). The first 8 pulses transmit the byte (MSB first), the ninth pulse is used by the receiver to acknowledge.

Figure 96. S-Bus / I^{2}C-bus Peripheral Compatibility without S-Bus Chip Select

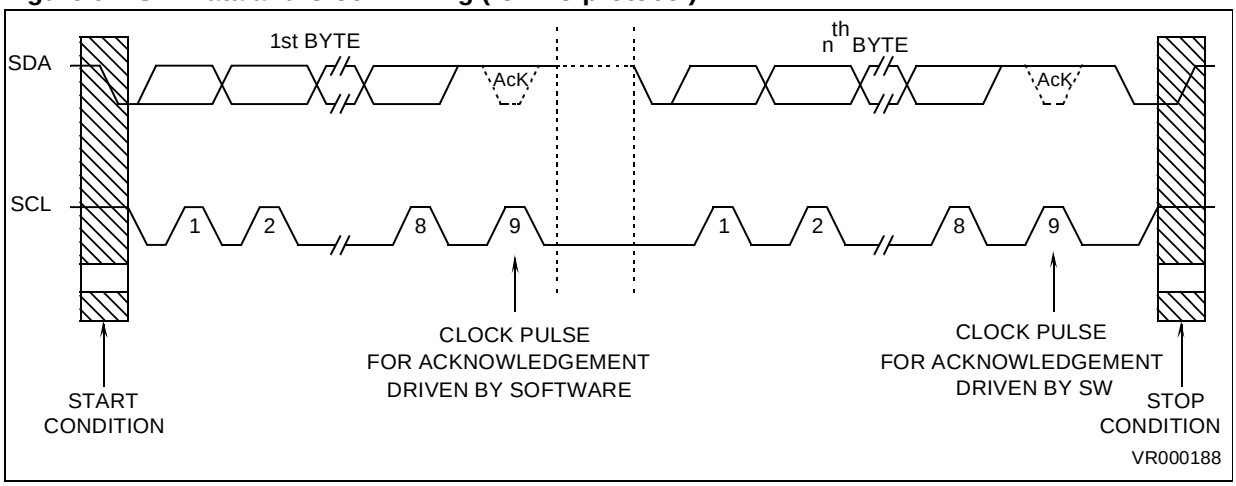


SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 32. Typical I^2C-bus Sequences

Phase	Software	Hardware	Notes
INITIALIZE	SPICR.CPOL, CPHA = 0, 0 SPICR.SPEN = 0 SPICR.BMS = 1 SCK pin set as AF output SDI pin set as input Set SDO port bit to 1	SCK, SDO in HI-Z SCL, SDA = 1, 1	Set polarity and phase SPI disable START/STOP interrupt Enable
START	SDO pin set as output Open Drain Set SDO port bit to 0	SDA = 0, SCL = 1 interrupt request	START condition receiver START detection
TRANSMISSION	SPICR.SPEN = 1 SDO pin as Alternate Function output load data into SPIDR	SCL = 0 Start transmission Interrupt request at end of byte transmission	Managed by interrupt routine load FFh when receiving end of transmission detection
ACKNOWLEDGE	SPICR.SPEN = 0 Poll SDA line Set SDA line SPICR.SPEN = 1	SCK, SDO in HI-Z SCL, SDA = 1 SCL = 0	SPI disable only if transmitting only if receiving only if transmitting
STOP	SDO pin set as output Open Drain SPICR.SPEN = 0 Set SDO port bit to 1	SDA = 1 interrupt request	STOP condition

Figure 97. SPI Data and Clock Timing (for I2C protocol)



SERIAL PERIPHERAL INTERFACE (Cont'd)

The data on the SDA line is sampled on the low to high transition of the SCL line.

SPI working with an I^2C-bus

To use the SPI with the I^2C-bus protocol, the SCK line is used as SCL; the SDI and SDO lines, externally wire-ORed, are used as SDA. All output pins must be configured as open drain (see Figure 95).

Table 32. illustrates the typical I^2C-bus sequence, comprising 5 phases: Initialization, Start, Transmission, Acknowledge and Stop. It should be noted that only the first 8 bits are handled by the SPI peripheral; the ACKNOWLEDGE bit must be managed by software, by polling or forcing the SCL and SDO lines via the corresponding I/O port bits.

During the transmission phase, the following I^2C-bus features are also supported by hardware.

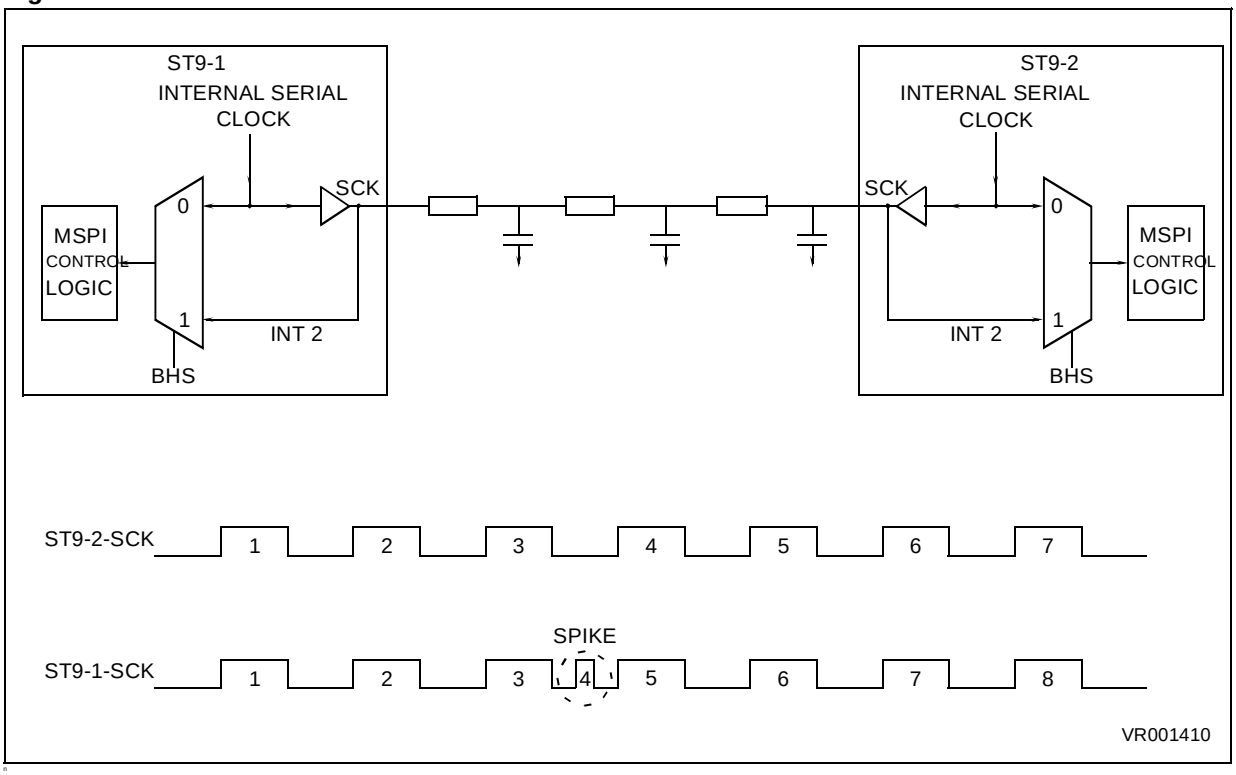
Clock Synchronization

In a multimaster I^2C-bus system, when several masters generate their own clock, synchronization is required. The first master which releases the SCL line stops internal counting, restarting only when the SCL line goes high (released by all the other masters). In this manner, devices using different clock sources and different frequencies can be interfaced.

Arbitration Lost

When several masters are sending data on the SDA line, the following takes place: if the transmitter sends a "1" and the SDA line is forced low by another device, the ARB flag (SPICR.5) is set and the SDO buffer is disabled (ARB is reset and the SDO buffer is enabled when SPIDR is written to again). When BMS is set, the peripheral clock is supplied through the INT2 line by the external clock line (SCL). Due to potential noise spikes (which must last longer than one INTCLK period to be detected), RX or TX may gain a clock pulse. Referring to Figure 97, if device ST9-1 detects a noise spike and therefore gains a clock pulse, it will stop its transmission early and hold the clock line low, causing device ST9-2 to freeze on the 7th bit. To exit and recover from this condition, the BMS bit must be reset; this will cause the SPI logic to be reset, thus aborting the current transmission. An End of Transmission interrupt is generated following this reset sequence.

Figure 98. SPI Arbitration



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.9.7 S-Bus Interface

The S-bus is a three-wire bidirectional data-bus, possessing functional features similar to the I^2C-bus. As opposed to the I^2C-bus, the Start/Stop conditions are determined by encoding the information on 3 wires rather than on 2, as shown in Figure 99. The additional line is referred as SEN.

Figure 99. Mixed S-bus and I^2C-bus System

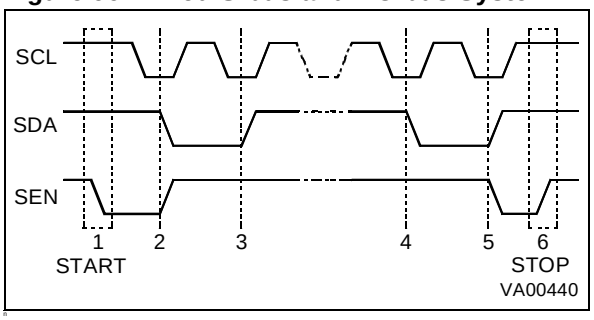
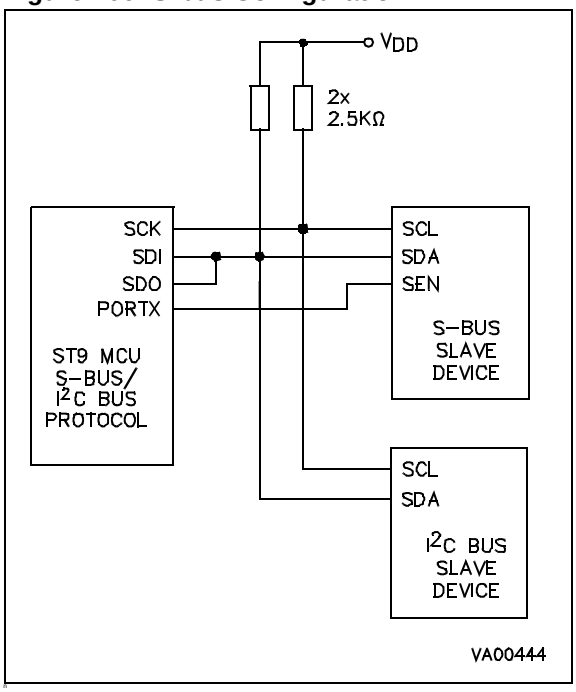


Figure 100. S-bus Configuration



SPI Working with S-bus

The S-bus protocol uses the same pin configuration as the I^2C-bus for generating the SCL and SDA lines. The additional SEN line is managed through a standard ST9 I/O port line, under software control (see Figure 95).

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.9.8 IM-bus Interface

The IM-bus features a bidirectional data line and a clock line; in addition, it requires an IDENT line to distinguish an address byte from a data byte (Figure 101). Unlike the I^2C-bus protocol, the IM-bus protocol sends the least significant bit first; this requires a software routine which reverses the bit order before sending, and after receiving, a data byte. Figure 100 shows the connections between an IM-bus peripheral and an ST9 SPI. The SDO and SDI pins are connected to the bidirectional data pin of the peripheral device. The SDO alternate function is configured as Open-Drain (external 2.5KΩ pull-up resistors are required).

With this type of configuration, data is sent to the peripheral by writing the data byte to the SPIDR register. To receive data from the peripheral, the user should write FFh to the SPIDR register, in order to generate the shift clock pulses. As the SDO line is set to the Open-Drain configuration, the incoming data bits that are set to "1" do not affect the SDO/SDI line status (which defaults to a high level due to the FFh value in the transmit register), while incoming bits that are set to "0" pull the input line low.

In software it is necessary to initialise the ST9 SPI by setting both CPOL and CPHA to "1". By using a general purpose I/O as the IDENT line, and forcing it to a logical "0" when writing to the SPIDR register, an address is sent (or read). Then, by setting this bit to "1" and writing to SPIDR, data is sent to the peripheral. When all the address and data pairs are sent, it is necessary to drive the IDENT line low and high to create a short pulse. This will generate the stop condition.

Figure 101. ST9 and IM-bus Peripheral

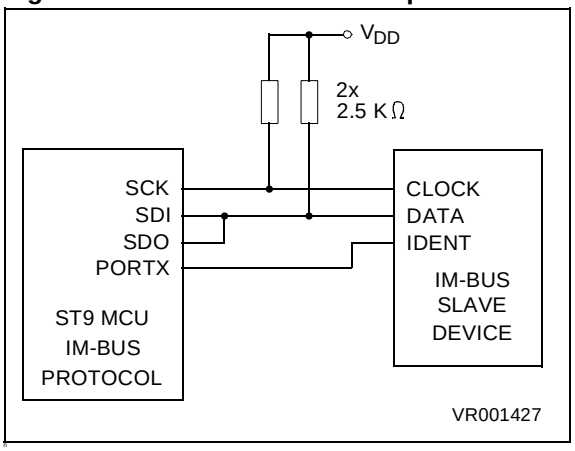
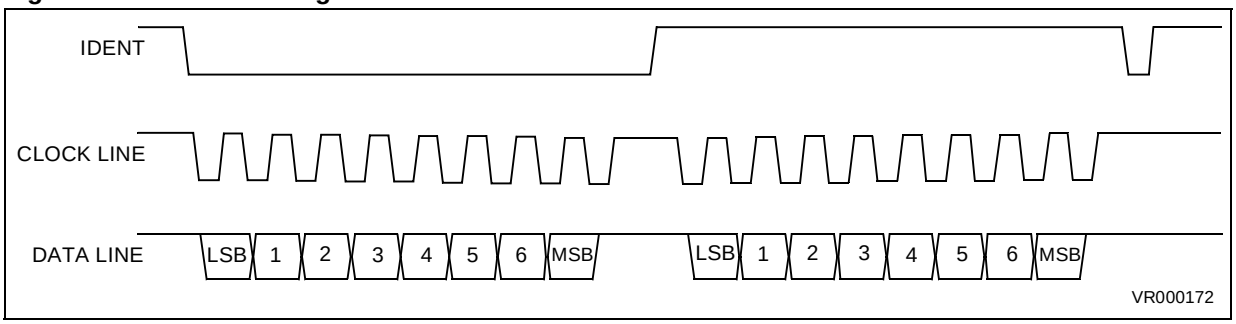


Figure 102. IM bus Timing



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.9.9 Register Description

It is possible to have up to 3 independent SPIs in the same device (refer to the device block diagram). In this case they are named SPI0 thru SPI2. If the device has one SPI converter it uses the register adresses of SPI0. The register map is the following:

Register	SPIn	Page
SPIDR R253	SPI0	0
SPICR R254	SPI0	0
SPIDR1 R253	SPI1	7
SPICR1 R254	SPI1	7
SPIDR2 R245	SPI2	7
SPICR2 R246	SPI2	7

Note: In the register description on the following pages, register and page numbers are given using the example of SPI0.

SPI DATA REGISTER (SPIDR)
R253 - Read/Write
Register Page: 0
Reset Value: undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[0:7]**: *SPI Data*.
This register contains the data transmitted and received by the SPI. Data is transmitted bit 7 first, and incoming data is received into bit 0. Transmission is started by writing to this register.

Note: SPIDR state remains undefined until the end of transmission of the first byte.

SPI CONTROL REGISTER (SPICR)
R254 **-** Read/Write
Register Page: 0
Reset Value: 0000 0000 (00h)

7							0
SPEN	BMS	ARB	BUSY	CPOL	CPHA	SPR1	SPR0

Bit 7 = **SPEN**: *Serial Peripheral Enable*.
0: SCK and SDO are kept tristate.
1: Both alternate functions SCK and SDO are enabled.

Note: furthermore, SPEN (together with the BMS bit) affects the selection of the source for interrupt channel B0. Transmission starts when data is written to the SPIDR Register.

Bit 6 = **BMS**: *S-bus/I^{2}C-bus Mode Selector*.
0: Perform a re-initialisation of the SPI logic, thus allowing recovery procedures after a RX/TX failure.
1: Enable S-bus/I^{2}C-bus arbitration, clock synchronization and Start/ Stop detection (SPI used in an S-bus/I^{2}C-bus protocol).

Note: when the BMS bit is reset, it affects (together with the SPEN bit) the selection of the source for interrupt channel B0.

Bit 5 = **ARB**: *Arbitration flag bit.*
This bit is set by hardware and can be reset by software.
0: S-bus/I^{2}C-bus stop condition is detected.
1: Arbitration lost by the SPI in S-bus/I^{2}C-bus mode.

Note: when ARB is set automatically, the SDO pin is set to a high value until a write instruction on SPIDR is performed.

Bit 4 = **BUSY**: *SPI Busy Flag*.
This bit is set by hardware. It allows the user to monitor the SPI status by polling its value.
0: No transmission in progress.
1: Transmission in progress.

Bit 3 = **CPOL**: *Transmission Clock Polarity*.
CPOL controls the normal or steady state value of the clock when data is *not* being transferred. Please refer to the following table and to Figure 102 to see this bit action (together with the CPHA bit).

Note: As the SCK line is held in a high impedance state when the SPI is disabled (SPEN = "0"), the SCK pin must be connected to V_{SS} or to V_{CC} through a resistor, depending on the CPOL state. Polarity should be set during the initialisation routine, in accordance with the setting of all peripherals, and should not be changed during program execution.

SERIAL PERIPHERAL INTERFACE (Cont'd)

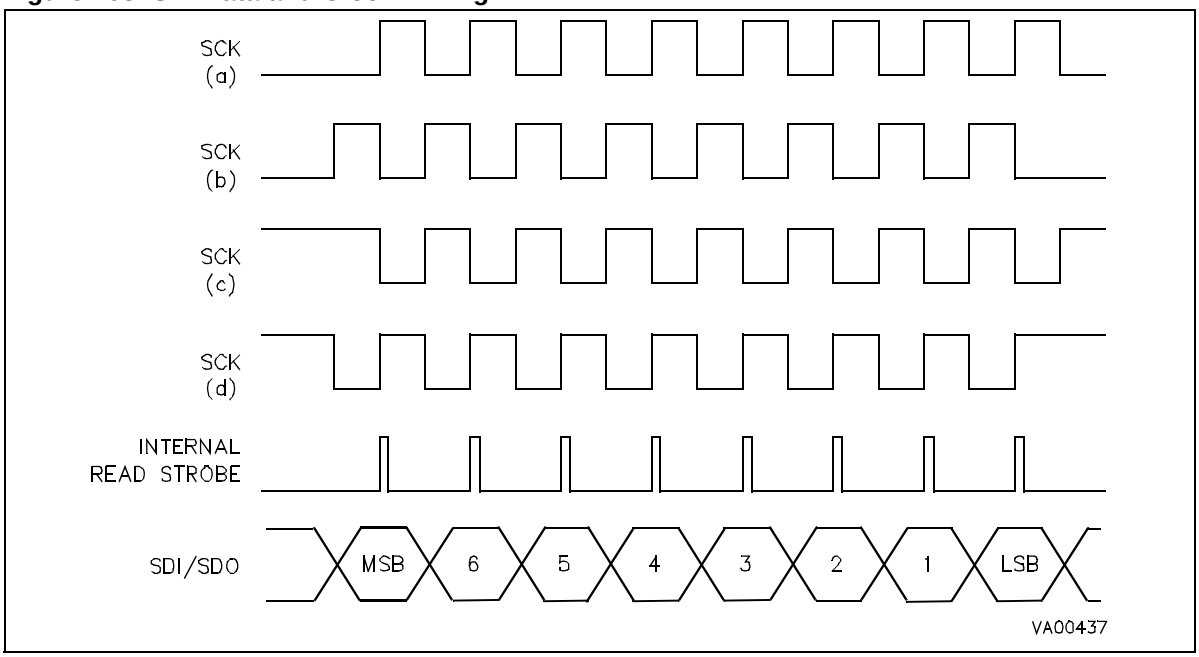
Bit 2 = **CPHA**: *Transmission Clock Phase.*
CPHA controls the relationship between the data on the SDI and SDO pins, and the clock signal on the SCK pin. The CPHA bit selects the clock edge used to capture data. It has its greatest impact on the first bit transmitted (MSB), because it does (or does not) allow a clock transition before the first data capture edge. Figure 102 shows the relationship between CPHA, CPOL and SCK, and indicates active clock edges and strobe times.

CPOL	CPHA	SCK (in Figure 102)
0	0	(a)
0	1	(b)
1	0	(c)
1	1	(d)

Bit 1:0 = **SPR[1:0]:** *SPI Rate.*
These two bits select one (of four) baud rates, to be used as SCK.

SPR1	SPR0	Clock Divider	SCK Frequency (@ INTCLK = 24MHz)	
0	0	8	3000kHz	(T = 0.33µs)
0	1	16	1500kHz	(T = 0.67µs)
1	0	128	187.5kHz	(T = 5.33µs)
1	1	256	93.75kHz	(T = 10.66µs)

Figure 103. SPI Data and Clock Timing



7.10 TWO-CHANNEL I2C BUS INTERFACE (I2C)

7.10.1 Introduction

The I2C Bus Master/Slave Interface supports up to 2 serial I2C buses used for communication with various external devices. It meets all of the requirements of the I2C bus specification (except extended 10-bit addressing compatibility for slave operation and CBUS compatibility).

7.10.1.1 General Features

– Conversion of internal 8-bit parallel data to/from external I2C bus serial data
– Realtime interrupt generation and handling
– Software selectable operation one of two I2C buses
– Software selectable acknowledge bit generation
– Internal general reset
– 8-bit data read/write register
– 8-bit control register,
– 8-bit status register,
– Operates by default in slave mode and is automatically switched to master mode by loading the ‘data write’ register when the bus is idle.

7.10.1.2 Master Operation

– 4-bit Frequency Control register to select 1 of 16 clock frequencies for the SCL line ranging from 20 kHz to 800 kHz derived from a 4MHz crystal clock
– Compatible with standard 7 or extended 10-bit address protocol
– Handles stretching of SCL bus clock pulses by slaves without restrictions
– Bus arbitration with arbitration loss detection in multimaster environment
– Bus error detection
– Optional push-pull bus drive capability for faster communication

7.10.1.3 Slave Operation

– 7-bit address register (cannot be assigned a 10-bit address)
– The first SCL clock pulse in every data byte is stretched until the MCU has finished processing the previously received byte
– Bus error detection
– Optional general call detection
– Operates optionally as Bus Monitor without interfering in any way with bus traffic
– Setup time for any first transmitted data bit can be adjusted

I²C BUS INTERFACE (Cont'd)

7.10.2 General Description

In addition to receiving and transmitting data, this interface convert them from serial to parallel format and vice versa. The interface is connected, through a multiplexer, to one I²C bus among 2 by a data pin, SDAx, and by a clock pin, SCLx, where x range value is 1 to 2.
It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

7.10.2.1 Mode Selection

The interface can operate in the four following modes:

– Slave transmitter/receiver

– Master transmitter/receiver

By default, it operates in slave mode.

The interface automatically switches from inactive slave to master after it generates a START condition and from master to inactive slave in case of arbitration loss or a STOP generation, this allows Multi-Master capability.

7.10.2.2 Communication Flow

In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Start condition is automatically generated by the interface when the data register is loaded by the slave address (see register description for further details). Stop condition is generated in master mode by writing by software in the control register.

In Slave mode, the interface is capable of recognising its own address (7-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

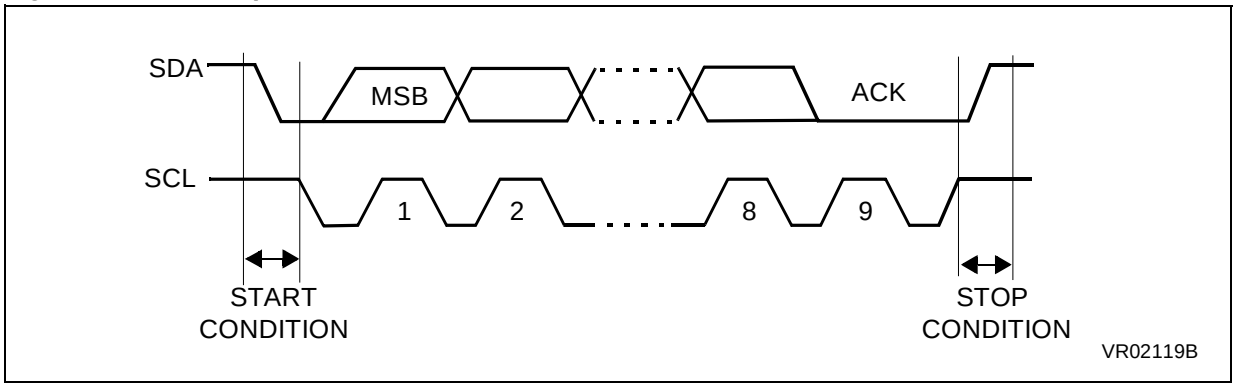
Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition is the address byte; it is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to Figure 1.

The acknowledge may be enabled and disabled by software.

The speed of the I²C interface may be selected between Standard (15.625 - 100 kHz), Fast I²C (100- 400 kHz), or Extended I²C (500 - 800 kHz).

Figure 104. I²C bus protocol



I²C BUS INTERFACE (Cont'd)

7.10.3 Functional Description

Refer to Section 0.1.6 for the bit definitions.

Figure 2 gives the block diagram of the cell.

By default, the I²C interface is in inactive slave mode, except when it initiates a transmit or receive sequence.

After the microcontroller power-on reset state, the I²C interface is in reset state until the CLEAR bit (I2CCTR register) is reset.

7.10.3.1 Configuring the interface

Before using the I²C interface, configure it as follows.

If it is to be used in slave mode, write the address assigned to the interface in the I2COAR register.

If it is to be used in master mode, write the SCL clock frequency in the I2CFQR register.

Then, select one of the two buses available and configure the corresponding pins to the alternate function (refer to the I/O port chapter).

Depending on your application, you may use the advanced features (see the UNPROC and UNEXP bits of the I2CSTR2 register) by setting the AFEN bit of the I2CCTR register.

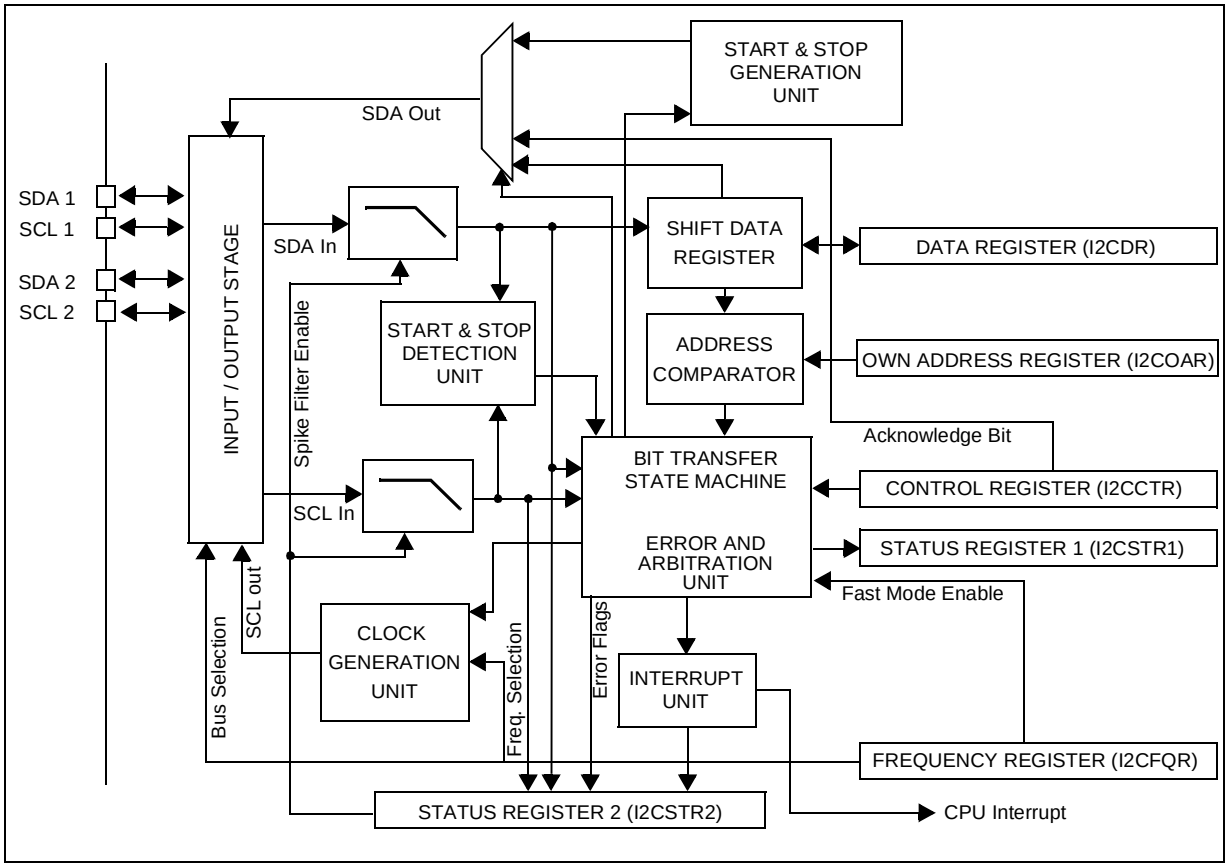
You may also optionally set the RSRT and STOP bits of the I2CCTR register.

You can enable the interrupt on stop condition and the Spike filter by setting the ISCEN and SFEN bits of the I2CSTR1 register.

If you want to use the monitor feature, then set the MONITOR bit in the I2CCTR register.

In all cases reset the CLEAR bit of the I2CCTR register to enable the I²C interface.

Figure 105. I²C interface block diagram



I²C BUS INTERFACE (Cont'd)

7.10.3.2 Slave Mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with:

– The 7 MSB of the interface address (see I2COAR register) if the ADR0 bit = 0

– The 4 MSB of the interface address (see I2COAR register) if the ADR0 bit = 1

– The General Call address

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:

– Acknowledge pulse if the GENC_ACK bit (I2CCTR register) is set and a general call is detected, or if the SEND_ACK bit (I2CCTR register) is reset and "normal" address if detected.

– An interrupt is generated and the INT bit of the I2CSTR2 register is set.

Then check the I2CSTR1 register to know the interface status:

– Read the FIRST bit of the I2CSTR1 register to know whether the byte stored in the I2CDR register is the address (first byte transferred in an I2C transaction) or a data.

– If the GEN_CALL bit is set, a general call has been requested by a master.

– If the ACT_SLV bit is set and the READ bit is set, the interface is an active slave transmitter, else, if the ACT_SLV bit is set and the READ bit is reset, the interface is an active slave receiver.

Slave receiver

After the address, the slave receives bytes from the SDA line into the I2CDR register via the internal shift register. After each received byte the interface generates in sequence:

– Acknowledge pulse according to the SEND_ACK bit value

– An interrupt is generated and the INT bit of the I2CSTR2 register is set.

Using the FIRST bit of the I2CSTR1 register, you know whether the byte stored in the I2CDR register is the address (first byte transferred in an I2C transaction) or data.

Slave Transmitter

Following the address reception, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave writes in the I2CDR register the data to send on the SDA bus.

When the acknowledge pulse is received:

– an interrupt is generated and the INT bit of the I2CSTR2 register is set.

Then you need to check the ACK_BIT of the I2CSTR2 register to know whether the last byte has been acknowledged or not. If some data have to be sent again, write the value in the OSDDR register.

Closing a slave communication

The I2C interface returns to inactive slave state as soon as a stop condition has been detected.

If the ISCEN bit of the I2CSTR2 is set, an interrupt is generated on detecting the stop condition, allowing the user to know if the transaction was successful by checking the ERROR and ACTIVE flags of the I2CSTR1 register.

7.10.3.3 Master Mode

To switch from default inactive slave mode to Master mode: load a slave address in the I2CDR register.

If the bus is free (ACTIVE bit of the I2CSTR2 register reset), then the I²C interface automatically generates a start condition followed by the I2CDR byte.

Then, on the 9th clock pulse, an interrupt is generated and the INT bit of the I2CSTR2 register is set.

Check the ACK_BIT bit of the I2CSTR1 register to know whether the slave address has been acknowledged or not, in order to manage the transaction.

If needed, generate a stop condition on the bus with the STOP bit of the I2CSTR1 register.

Note: If the RSRT bit of the I2CCTR register is set, the master will generate a repeated start sequence as soon as a new byte is loaded in the I2CDR register.

I²C BUS INTERFACE (Cont'd)

Master Receiver

Following the address transmission and acknowledgment, the master receives bytes from the SDA line into the I2CDR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse according to the SEND_ACK bit value,

– An interrupt is generated and the INT bit of the I2CSTR2 register is set.

Then read the I2CDR register to store the transmitted data.

Note: In order to generate the non-acknowledge pulse after the last received data byte, the SEND_ACK bit must be set just before reading the second last data byte.

Master Transmitter

Following the address transmission and acknowledgment, the master sends bytes from the I2CDR register to the SDA line via the internal shift register.

When the acknowledge bit is received, the interface generates an interrupt and sets the INT bit of the I2CSTR2 register.

The user can check the ACK_BIT bit of the I2CSTR1 register in order to handle the transaction properly.

Closing a master communication

The master interface will generate a stop condition on the bus when the user sets the STOP bit of the I2CSTR1 register.

7.10.4 Interrupt Handling

To acknowledge interrupts generated by the I2C interface, software must write any value in the I2CDR register before leaving the I2C interrupt subroutine. This is necessary in all modes including slave or master receiver mode.

7.10.5 Error Cases

Each time an error occurs, an interrupt is generated. Then by checking the following bits, the user can identify the problem:

– If the **ERROR** bit in the I2CSTR1 register is set, an illegal start or stop condition has been detected.
If the AFEN bit in the I2CCTR register is set, the UNPROC, UNEXP, and MISP bits of the I2CSTR2 register indicate what kind of illegal condition has been detected.

– If the **ARB_LOST** bit, in the I2CSTR1 register is set, an arbitration lost occurred on the bus.

Note: the ERROR bit has higher priority than the ARB_LOST bit, so if ERROR is set, ARB_LOST has to be ignored.

I²C BUS INTERFACE (Cont'd)

7.10.6 Register Description

OWN ADDRESS REGISTER (I2COAR)
R240 - Read/Write
Register Page: 44
Reset Value: 0000 0000(00h)

7							0
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

Bit 7:1 = **ADR[7:1]** *Interface Slave Address*
These bits are the 7 most significant bits of the 8-bit address assigned to interface when it works in slave mode.

Bit 0 = **ADR0** *Address match bit*
This bit selects when the I²C interface becomes active if it works in slave mode, and if its slave address is transmitted on the bus.
0: The interface becomes an active slave when the 7 most significant bits ADR[7:1] match the address transmitted by a master.
1: The interface becomes an active slave when only the 4 most significant bits ADR[7:4] match the address transmitted by a master. This feature allows the master to send data simultaneously to up to 8 slaves with identical ADR(7:4).

I²C BUS INTERFACE (Cont'd)

FREQUENCY REGISTER (I2CFQR)
R241 - Read/Write
Register Page: 44
Reset Value: 0000 0000(00h)

7							0
BUS_S0	BUS_S1	FMEN	PP_DRV	Q3	Q2	Q1	Q0

Bits 7:6 = **BUS_S[1:0]** *I²C BUS Selection bits*
These bits connect the I²C interface to one of the two possible buses as described in Table 1.

Table 33. I²C bus selection

BUS_S1	BUS_S0	Selected Bus
0	0	SCL1/SDA1
0	1	SCL2/SDA2

Bit 5 = **FMEN** *Fast Mode Enable bit*
This bit enables or disables the fast mode for the SCL bus frequency.
0: Standard Mode (up to 100 kHz).
1: Fast Mode (over 100 kHz)

Bit 4 = **PP_DRV** *Push-Pull Drive mode bit*
This bit determines if the master drives the SCL/SDA buses in push-pull mode or in normal mode. This allows the master to send data to the slave at a faster speed.
0: The push-pull drive mode is disabled
1: The push-pull drive mode is enabled. All "normal" bus frequencies are doubled with the only exception that the push-pull drive mode is automatically disabled when Q[3:0]=1110 or Q[3:0]=1111 to yield an SCL frequency of 500 kHz or 800 kHz. Refer to Q[3:0] bit description.

Note: The master automatically switches temporarily to normal bus driving mode with active pull-up disabled and SCL frequency reduced by factor of 2 when receiving acknowledges or data from the addressed slave.

Bit 3:0 = **Q[3:0]** *SCL clock frequency bits*
These bits select the SCL clock frequency when the interface works in master mode. In slave transmitter mode, they can be used to adjust the setting up time between the first data byte and the clock. Refer to Table 2.

In push-pull mode, the frequency values presented in the following table correspond to an approximate frequency assuming that :

– the first data bit is transferred at a lower frequency (clock stretching capability),
– the acknowledge bit is transferred at the slave speed without push-pull mode,
– other data bits are transferred with a real period 250 ns shorter than the values indicated in this table.

Using the spike filter will add an internal delay acting as a period increase by 250-ns steps.

Table 34. SCL Clock Frequency Selection

	SCL BUS FREQUENCY (in kHz)	
Q[3:0]	PP_DRV = 0 SCL max Frequency	PP_DRV = 1 SCL Frequency (kHz) (period: +0/-250ns
0000	20.10	40.40
0001	30.53	61.54
0010	40.40	81.63
0011	50.63	102.56
0100	63.49	129.03
0101	72.73	148.15
0110	85.11	173.91
0111	102.56	210.53
1000	129.03	266.67
1001	173.91	363.64
1010	210.53	444.44
1011	266.67	444.44
1100	400.00	571.43
1101	444.44	666.67
1110	444.44*	666.67
1111	666.67*	666.67

* These values are not covered by the Philips I²C specification

Notes:
– The maximum allowed frequency depends on the state of the FMEN control bit (If PP_DRV=0, standard mode: 100 KHz; fast mode: 666.6 kHz)
– All frequency values depend on the bus line load (except push-pull mode).
– All above values are obtained with loads corresponding to a rise time from 0 to 250 ns.
– Any higher rise time (especially in standard mode) will increase the period of the bus line frequency by 250-ns steps.

I²C BUS INTERFACE (Cont'd)

CONTROL REGISTER (I2CCTR)
R242 - Read/Write
Register Page: 44
Reset Value: 0000 0001(01h)

7							0
AFEN	RTI	GENC_ ACK	SEND_ ACK	MONI TOR	RSRT	STOP	CLEAR

Bit 7 = **AFEN** *Advanced Features Enable bit*
This bit enables or disables the unexpected & unprocessed error detection. Refer to the description of the UNPROC and UNEXP bits in the I2CSTR2 register.
0: Advanced features disabled
1: Advanced features enabled

Bit 6 = **RTI** *Return To Inactive state bit*
This bit determines the interface status after an interrupt is processed (either after a complete transfer or an error occured).
0: The interface keeps its active state
1: The interface (master or slave) returns to the inactive slave state

Note: The state of the Active Flag (I2CSTR1.0) is maintained.The RTI bit is automatically cleared.

Bit 5 = **GENC_ACK** *General Call Acknowledge bit*
This bit determines the response of the I²C interface when a general call is detected on the bus.
0: The interface will acknowledge the reception of a 'General Call' immediately after receiving the address 00h. An interrupt is generated at the end of the acknowledge interval that follows the address.
1: The interface will not acknowledge a 'General Call' and does not generate an interrupt, i.e. the interface will remain an inactive slave.

Bit 4 = **SEND_ACK** *Send Acknowledge bit*
This bit is set by software to define if the acknowledge bit is placed on the bus when the interface is operating as a master receiver, active slave receiver or an active slave.
0: An inactive interface will acknowledge the reception of its address and switch to active slave mode.
1: The interface will not acknowledge the reception of its address and remains inactive.

Note: The interface operating as a master slave receiver is free to acknowledge or not all data bytes. In a normal I²C transaction, it acknowledges all data bytes except the last received from a slave/master transmitter.
SEND_ACK should be programmed before receiving the relevant byte (data or address).

Bit 3 = **MONITOR** *Bus Monitor mode bit*
This bit determines if the interface acts as a bus monitor or not.
0: The bus monitor mode is disabled.
1: The interface behaves as a bus monitor. The interface becomes a slave regardless of the address received, but neither the address or the following data is acknowledged (this is equivalent to SEND_ACK=1). If a read address is received, the high state of the least significant bit of this address is suppressed inside the interface and all data bytes are processed by the MCU as received data.

Bit 2 = **RSRT** *Repeated Start bit*
This bit determines if the interface generates automatically a repeated start condition on the I²C bus (in master mode) as soon as a new byte is ready to be send.
0: Repeated start disabled
1: Repeated start enabled

Note: This bit is automatically cleared.

Bit 1 = **STOP** *STOP condition generation bit*
When working in master mode, this bit enables or disables a STOP condition generation on the I²C bus.
0: No Stop condition is generated
1: The master will generate a stop condition to terminate the bus transaction. The master will automatically revert to an inactive slave and the STOP bit will be cleared.

Bit 0 = **CLEAR** *Clear interface bit*
This bit enables or disables the I²C interface.
0: The interface is enabled
1: A general reset is generated. The interface becomes an inactive slave and the SCL and SDA buses drive signals are removed. The system is kept in reset state until the CLEAR bit is written to "0".

Note: The CLEAR bit is "1" (i.e. the interface is disabled) when exiting from the MCUs power-on reset state.

I²C BUS INTERFACE (Cont'd)

DATA REGISTER (I2CDR)
R243 - Read/Write
Register Page: 44
Reset Value: 0000 0000(00h)

7							0
SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1

Bit 7:0 = **SR[8:1]** *address or data byte*
These bits contains the address or data byte loaded by software for sending on the I²C bus, and also the address or data byte received on the bus to be read by software.
When read, this register reflects the last byte which has been transferred on the bus. Reading this register is equivalent to reading the shift register of the interface.
When written, the contents of this register will be transferred into the shift register of the interface.

STATUS REGISTER 2 (I2CSTR2)
R244 - Read/Write (Bit 7:6),Read Only (Bit 5:0)
Register Page: 44
Reset Value: 0000 0000(00h)

7							0
ISCEN	SFEN	SCLIN	SDAIN	INT	UNPROC	UNEXP	MISP

Bit 7 = **ISCEN** *Interrupt on Stop Condition Enable bit*
This bit determines if an interrupt is generated as soon as a stop condition has been detected on the bus.
0: No interrupt generated on a bus stop condition
1: An interrupt is generated on a bus stop condition.

Note: When the interface is involved in a transaction, checking the ERROR status flag related to the error detection allows to determine if the transaction has been successfully completed. This interrupt can be useful for an interface waiting for a "bus free" condition in order to become a master as soon as possible. Checking the ACTIVE bit (in the I2CSTR1 register) allows to correctly identify an interrupt generated by a stop condition.

Bit 6 = **SFEN** *Spike Filter Enable bit*
This bit enables or disables the spike filters on the SDAx and SCLx inputs (x is 1 to 2).
0: spike filters disabled
1: spike filters enabled

Note: The length of a pulse identified as a spike depends on the CPUCLK frequency used (CPUCLK frequencies from 10 to 20 Mhz allow to filter pulses smaller than 100 to 40 ns).

Bit 5 = **SCLIN** *SCL Input status bit*
This read-only bit describes the current logic state on the SCL bus.
It can be used to sample the signal on a newly selected SCL bus for a quick determination concerning the bus use and the bus clock frequency.

Bit 4 = **SDAIN** *SDA Input status bit*
This read-only bit describes the current logic state on the SDA bus.
It can be used to sample the signal on a newly selected SDA bus for a quick determination of the state of this bus, prior starting a transaction.

Bit 3 = **INT** *Interrupt status bit*

This (read-only) bit indicates if an event has occurred.
0: No interrupt requested or an interrupt resulting from a stop condition occurred.
1: The interface enters an interrupt state resulting from any error (bus error or arbitration loss) or any byte transfer completed.

Bit 2 = **UNPROC** *Unprocessed flag bit*
This bit is useful in a multimaster mode system, to solve conflicts between a "Repeated Start" or a "Stop" condition and any bit of an address or data byte from other concurrent masters.
0: No error occurred.
1: A master interface tried to generate a "Repeated Start" or a "Stop" condition, which never occurred.

Note: If this bit is set, it will automatically activate the ERROR bit.

Note: This bit is only valid when the Advanced Features Enable bit AFEN is set in the I2CCTR register.

I²C BUS INTERFACE (Cont'd)

Bit 1 = **UNEXP** *Unexpected flag bit*
This bit is useful for error detection in a multimaster mode system, when a master is continuing its transaction while an other concurrent master wants to finish or restart a transaction by sending a "Start" or a "Stop" condition.
Together with the MISP bit, it covers all possible cases, where unexpected "Start" or "Stop" conditions occur, while the interface is a master.
0: No Unexpected error detected
1: A master interface receives a "Start" or a "Stop" condition, while sending the first bit of a data byte.

Notes:

– If this bit is set, it will automatically activate the ERROR bit.

– This bit is only valid when the Advanced Features Enable bit AFEN is set.

Bit 0 = **MISP** *Misplaced flag bit*
This bit indicates if the interface has received a misplaced "Start" or "Stop" condition during address transfer or any data byte transfer (besides first data bit). This error detection is also activated during the acknowledge bit transfer.
Together with the UNEXP bit, it covers all possible cases, where unexpected "Start" or "Stop" conditions occur, while the interface is a master.
0: No misplaced "start" or "stop" condition has been detected
1: A misplaced "Start" or "Stop" condition has been received.

Note: If this bit is set, it will automatically activate the ERROR bit.

STATUS REGISTER 1 (I2CSTR1)
R245 - Read Only
Register Page: 44
Reset Value: 0000 0000(00h)

7							0
ERROR	ARB_ LOST	READ	FIRST	GEN_ CALL	ACK_ BIT	ACT_ SLV	ACTIVE

Bit 7 = **ERROR** *ERROR detection bit*
This bit indicates if an error occurred on the bus or not.
0: No error detected
1: An error is detected. It is an illegal start or stop condition, i.e. a signal level transition occurs on the SDA bus during presence of a clock pulse on the SCL bus.
An interrupt is generated in this case. The interface stays in the error state until the error flag is reset by either a CLEAR operation, a STOP request or a "Return To Inactive State" operation.

Note: the ERROR bit has higher priority than the ARB_LOST bit (i.e. when ERROR=1, the value of ARB_LOST has to be ignored).

I²C BUS INTERFACE (Cont'd)

Bit 6 = **ARB_LOST** *Arbitration LOST detection bit*
This bit indicates if an arbitration lost occurred on the bus.
0: No arbitration lost occurred
1: An arbitration lost occurred. The bit is set when the interface operating as a master loses arbitration to another master on the bus.
If a loss of arbitration occurs during the address byte and if the interface has been addressed by the winning master (ACT_SLV=1), then the ARB_LOST flag is cleared by any "data load" operation into I2CDR.
In all other cases it is up to the user to return the interface into the status of an inactive slave via either a CLEAR operation, a "Return To Inactive State" operation or a STOP request.
If a loss of arbitration occurs, an interrupt is generated: when occurring during the address byte, the interrupt is generated at the end of the acknowledge bit; when occurring during a data byte, the interrupt is generated immediately.

Note: the ERROR bit has higher priority than the ARB_LOST bit (i.e. when ERROR=1, the value of ARB_LOST has to be ignored).

Bit 5 = **READ** *Read/write status bit*
This flag represents the state of the read/write bit of the address byte. It is updated either for a master or an active slave after the end of the address byte. It is cleared, when the interface returns to the inactive slave status (i.e. after the normal completion of a transaction, when exiting from any error state, ...).
0: Write operation
1: Read operation

Bit 4 = **FIRST** *transmission status bit*
This bit indicates if the byte transmitted on the bus is an address byte or a data byte.
0: The byte is a data byte
1: The byte is the address part of an I²C bus transaction.

Note: the FIRST bit is automatically cleared at the end of the interrupt, after the address, and when the interface returns into inactive slave state.

Bit 3 = **GEN_CALL** *General CALL status bit*
This bit indicates if a general call has been detected on the bus.
This bit is updated only if GENC_ACK=0 (see I2CCTR register for more details)
0: No general call detected, or GENC_ACK=1.
1: The general call address 00h has been recognized by the slave.

Note: This bit is cleared by hardware when the interface returns to the inactive slave status.

Bit 2 = **ACK_BIT** *Acknowledge BIT*
This bit reflects the logic level of the acknowledge bit detected at the end of the last byte (either address or data) transmitted on the I²C bus. It remains valid until the interface exits from the interrupt state.
0: Acknowledge detected
1: No acknowledge detected

Bit 1 = **ACT_SLV** *Active Slave status bit*
This bit indicates the slave status of the interface.
0: The interface is not working in slave mode. It may be inactive or in master mode (see the ACTIVE bit for more details).
1: The address assigned to the interface has been received on the bus and has been acknowledged by the interface (SEND_ACK=0).

Note: This bit is cleared when the interface returns into inactive slave state.

Bit 0 = **ACTIVE** *Interface Activity status bit*
This bit indicates if the interface is active or not.
0: The I²C interface is inactive
1: The Interface is active. The bit is set throughout the interval between a start condition and the first stop condition that follows on the I²C bus.

Note: It is reset by the CLEAR bit.

I²C BUS INTERFACE (Cont'd)

Table 35. I2C Interface Register Map and Reset Values

Address	Register Name	7	6	5	4	3	2	1	0
240	I2COAR Reset Value	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
241	I2CFQR Reset Value	BUS_S0	BUS_S1	FMEN	PP_DRV	Q3	Q2	Q1	Q0
242	I2CCTR Reset Value	AFEN	RTI	GENC_ACK	SEND_ACK	MONITOR	RSRT	STOP	CLEAR
243	I2CDR Reset Value	SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1
244	I2CSTR2 Reset Value	ISCEN	SFEN	SCLIN	SDAIN	INT	UNPROC	UNEXP	MISP
245	I2CSTR1 Reset Value	ERROR	ARB_LOST	READ	FIRST	GEN_ CALL	ACK_BIT	ACT_SLV	ACTIVE

7.11 A/D CONVERTER (A/D)

7.11.1 Introduction

The 8 bit Analog to Digital Converter uses a fully differential analog configuration for the best noise immunity and precision performance. The analog voltage references of the converter are connected to the internal AV_{DD} & AV_{SS} analog supply pins of the chip if they are available, otherwise to the ordinary V_{DD} and V_{SS} supply pins of the chip. The guaranteed accuracy depends on the device (see Electrical Characteristics). A fast Sample/Hold allows quick signal sampling for minimum warping effect and conversion error.

7.11.2 Main Features

- 8-bit resolution A/D Converter
- Single Conversion Time (including Sampling Time):
  - 138 internal system clock periods in slow mode (~5.6 µs @25 MHz internal system clock);
  - 78 INTCLK periods in fast mode (~6.5 µs @ 12 MHz internal system clock)
- Sample/Hold: Tsample=
  - 84 INTCLK periods in slow mode (~3.4 µs @25 MHz internal system clock)
  - 48 INTCLK periods in fast mode (~4 µs @12 MHz internal system clock)
- Up to 4 Analog Inputs (the number of inputs is device dependent, see device pinout)
- Single/Continuous Conversion Mode
- External/Internal source Trigger (Alternate synchronization)
- Power Down mode (Zero Power Consumption)
- 1 Control Logic Register
- 1 Data Register

7.11.3 General Description

Depending on the device, up to 8 analog inputs can be selected by software.

Different conversion modes are provided: single, continuous, or triggered. The continuous mode performs a continuous conversion flow of the selected channel, while in the single mode the selected channel is converted once and then the logic waits for a new hardware or software restart.

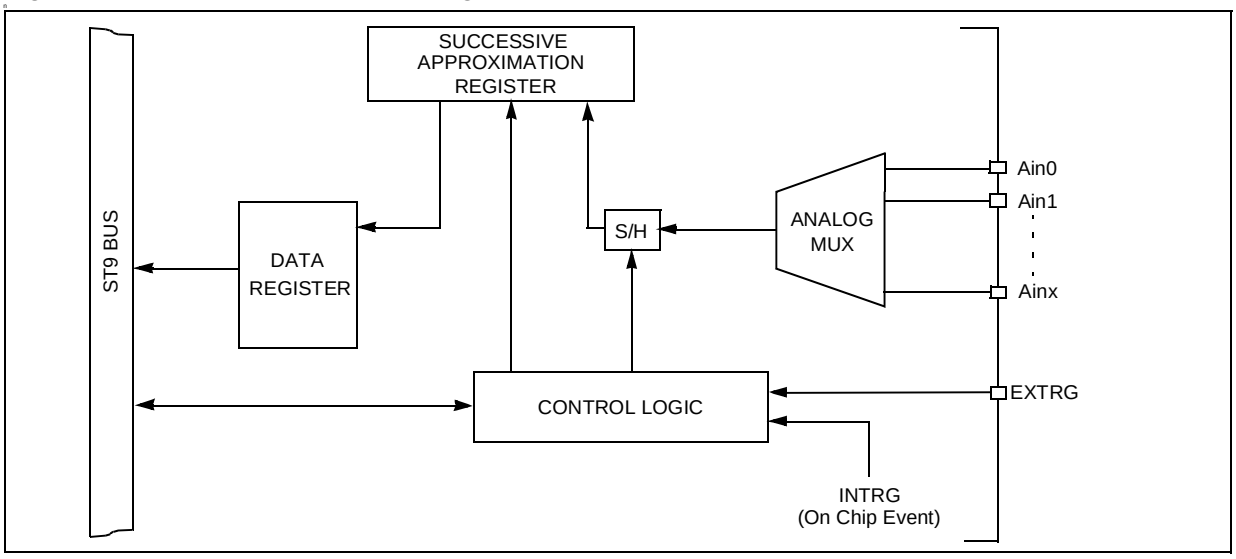
A data register (ADDTR) is available, mapped in page 62, allowing data storage (in single or continuous mode).

The start conversion event can be managed by software, writing the START/STOP bit of the Control Logic Register or by hardware using either:

– An external signal on the EXTRG triggered input (negative edge sensitive) connected as an Alternate Function to an I/O port bit

– An On Chip Event generated by another peripheral, such as the MFT (Multifunction Timer).

Figure 106. A/D Converter Block Diagram



A/D CONVERTER (Cont'd**)**

The conversion technique used is successive approximation, with AC coupled analog fully differential comparators blocks plus a Sample and Hold logic and a reference generator.

The internal reference (DAC) is based on the use of a binary-ratioed capacitor array. This technique allows the specified monotonicity (using the same ratioed capacitors as sampling capacitor). A Power Down programmable bit sets the A/D converter analog section to a zero consumption idle status.

7.11.3.1 Operating Modes

The two main operating modes, single and continuous, can be selected by writing 0 (reset value) or 1 into the CONT bit of the Control Logic Register.

Single Mode

In single mode (CONT=0 in ADCLR) the STR bit is forced to '0' after the end of channel i-th conversion; then the A/D waits for a new start event. This mode is useful when a set of signals must be sampled at a fixed frequency imposed by a Timer unit or an external generator (through the alternate synchronization feature). A simple software routine monitoring the STR bit can be used to save the current value before a new conversion ends (so to create a signal samples table within the internal memory or the Register File). Furthermore, if the R242.0 bit (register AD-INT, bit 0) is set, at the end of conversion a negative edge on the connected external interrupt channel (see Interrupts Chapter) is generated to allow the reading of the converted data by means of an interrupt routine.

Continuous Mode

In continuous mode (CONT=1 in ADCLR) a continuous conversion flow is entered by a start event on the selected channel until the STR bit is reset by software.

At the end of each conversion, the Data Register (ADCDR) content is updated with the last conversion result, while the former value is lost. When the conversion flow is stopped, an interrupt request is generated with the same modality previously described.

7.11.3.2 Alternate Synchronization

This feature is available in both single/continuous modes. The negative edge of external EXTRG signal or the occurrence of an on-chip event generated by another peripheral can be used to synchronize the conversion start with a trigger pulse. These events can be enabled or masked by programming the TRG bit in the ADCLR Register.

The effect of alternate synchronization is to set the STR bit, which is cleared by hardware at the end of each conversion in single mode. In continuous mode any trigger pulse following the first one will be ignored. The synchronization source must provide a pulse (1.5 internal system clock, 125ns @ 12 MHz internal clock) of minimum width, and a period greater (in single mode) than the conversion time (~6.5us @ 12 MHz internal clock). If a trigger occurs when the STR bit is still '1' (conversions still in progress), it is ignored (see Electrical Characteristics).

WARNING: If the EXTRG or INTRG signals are already active when TRG bit is set, the conversion starts immediately.

7.11.3.3 Power-Up Operations

Before enabling any A/D operation mode, set the POW bit of the ADCLR Register at least 60 µs before the first conversion starts to enable the biasing circuits inside the analog section of the converter. Clearing the POW bit is useful when the A/D is not used so reducing the total chip power consumption. This state is also the reset configuration and it is forced by hardware when the core is in HALT state (after a HALT instruction execution).

7.11.3.4 Register Mapping

It is possible to have two independent A/D converters in the same device. In this case they are named A/D 0 and A/D 1. If the device has one A/D converter it uses the register addresses of A/D 0. The register map is the following:

Register Address	ADn	Page 62 (3Eh)
F0h	A/D 0	ADDTR0
F1h	A/D 0	ADCLR0
F2h	A/D 0	ADINT0
F3-F7h	A/D 0	Reserved
F8h	A/D 1	ADDTR1
F9h	A/D 1	ADCLR1
FAh	A/D 1	ADINT1
FB-FFh	A/D 1	Reserved

If two A/D converters are present, the registers are renamed, adding the suffix 0 to the A/D 0 registers and 1 to the A/D 1 registers.

A/D CONVERTER (Cont'd**)**

7.11.4 Register Description

A/D CONTROL LOGIC REGISTER (ADCLR)
R241 - Read/Write
Register Page: 62
Reset value: 0000 0000 (00h)

7							0
C2	C1	C0	FS	TRG	POW	CONT	STR

This 8-bit register manages the A/D logic operations. Any write operation to it will cause the current conversion to be aborted and the logic to be re-initialized to the starting configuration.

Bits 7:5 = **C[2:0]**: *Channel Address.*
These bits are set and cleared by software. They select channel i conversion as follows:

C2	C1	C0	Channel Enabled
0	0	0	Channel 0
0	0	1	Channel 1
0	1	0	Channel 2
0	1	1	Channel 4
1	0	0	Channel 3

Bit 4 = **FS**: *Fast/Slow*.
This bit is set and cleared by software.
0: Fast mode. Single conversion time: 78 x INTCLK (5.75µs at INTCLK = 12 MHz)
1: Slow mode. Single conversion time: 138 x INTCLK (11.5µs at INTCLK = 12 MHz)

Note: Fast conversion mode is only allowed for internal speeds which do not exceed 12 MHz.

Bit 3 = **TRG**: *External/Internal Trigger Enable*.
This bit is set and cleared by software.
0: External/Internal Trigger disabled.
1: Either a negative (falling) edge on the EXTRG pin or an On Chip Event writes a "1" into the STR bit, enabling start of conversion.

Note: Triggering by on chip event is available on devices with the multifunction timer (MFT) peripheral.

Bit 2 = **POW**: *Power Enable*.
This bit is set and cleared by software.
0: Disables all power consuming logic.
1: Enables the A/D logic and analog circuitry.

Bit 1 = **CONT**: *Continuous/Single Mode Select*.
This bit it set and cleared by software.
0: Single mode: after the current conversion ends, the STR bit is reset by hardware and the converter logic is put in a wait status. To start another conversion, the STR bit has to be set by software or hardware.
1: Select Continuous Mode, a continuous flow of A/D conversions on the selected channel, starting when the STR bit is set.

Bit 0 = **STR**: *Start/Stop*.
This bit is set and cleared by software. It is also set by hardware when the A/D is synchronized with an external/internal trigger.
0: Stop conversion on channel i. An interrupt is generated if the STR was previously set and the AD-INT bit is set.
1: Start conversion on channel i

WARNING: When accessing this register, it is recommended to keep the related A/D interrupt channel masked or disabled to avoid spurious interrupt requests.

A/D CONVERTER (Cont'd**)**

A/D CHANNEL i DATA REGISTER (ADDTR)
R240 - Read/Write
Register Page: 62
Reset value: undefined

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

The result of the conversion of the selected channel is stored in the 8-bit ADDTR, which is reloaded with a new value every time a conversion ends.

Bits 7:0 = **R[7:0]**: *Channel i conversion result*.

A/D INTERRUPT REGISTER (ADINT)
Register Page: 62
R242 - Read/write
Reset value: 0000 0001 (01h)

7							0
0	0	0	0	0	0	0	AD-INT

Bits 7:1 = Reserved.

Bit 0 = **AD-INT**: *AD Converter Interrupt Enable*.

This bit is set and cleared by software. It allows the interrupt source to be switched between the A/D Converter and an external interrupt pin (See Interrupts chapter).
0: A/D Interrupt disabled. External pin selected as interrupt source.
1: A/D Interrupt enabled

7.12 VOLTAGE SYNTHESIS TUNING CONVERTER (VS)

7.12.1 Description

The on-chip Voltage Synthesis (VS) converter allows the generation of a tuning reference voltage in a TV set application. The peripheral is composed of a 14-bit counter that allows the conversion of the digital content in a tuning voltage, available at the VS output pin, by using PWM (Pulse Width Modulation) and BRM (Bit Rate Modulation) techniques. The 14-bit counter gives 16384 steps which allow a resolution of approximately 2 mV over a tuning voltage of 32 V. This corresponds to a tuning resolution of about 40 KHz per step in UHF band (the actual value will depend on the characteristics of the tuner).

The tuning word consists of a 14-bit word contained in the registers VSDR1 (R254) and VSDR2 (R255) both located in page 59.

Coarse tuning (PWM) is performed using the seven most significant bits. Fine tuning (BRM) is performed using the the seven least significant bits. With all “0”s loaded, the output is 0. As the tuning voltage increases from all “0”s, the number of pulses in one period increases to 128 with all pulses being the same width. For values larger than 128, the PWM takes over and the number of pulses in one period remains constant at 128, but the width changes. At the other end of the scale, when almost all “1”s are loaded, the pulses will start to link together and the number of pulses will decrease.

When all “1”s are loaded, the output will be almost 100% high but will have a low pulse (1/16384 of the high pulse).

7.12.2 Output Waveforms

Included inside the VS are the register latches, a reference counter, PWM and BRM control circuitry. The clock for the 14-bit reference counter is derived from the main system clock (referred to as INTCLK) after a division by 4. For example, using an internal 12 MHz on-chip clock (see Timing & Clock Controller chapter) leads to a 3 MHz input for the VS counter.

From the point of view of the circuit, the seven most significant bits control the coarse tuning, while the seven least significant bits control the fine tuning. From the application and software point of view, the 14 bits can be considered as one binary number.

As already mentioned the coarse tuning consists of a PWM signal with 128 steps: we can consider the fine tuning to cover 128 coarse tuning cycles.

The VS Tuning Converter is implemented with 2 separate outputs (VSO1 and VSO2) that can drive 2 separate Alternate Function outputs of 2 standard I/O port bits. A control bit allows you to choose which output is activated (only one output can be activated at a time).

When a VS output is not selected because the VS is disabled or because the second output is selected, it stays at a logical “one” level, allowing you to use the corresponding I/O port bit either as a normal I/O port bit or for a possible second Alternate Function output.

A second control bit allows the VS function to be started (or stopped) by software.

VOLTAGE SYNTHESIS (Cont'd)

PWM Generation

The counter increments continuously, clocked at INTCLK divided by 4. Whenever the 7 least significant bits of the counter overflow, the VS output is set.

The state of the PWM counter is continuously compared to the value programmed in the 7 most significant bits of the tuning word. When a match occurs, the output is reset thus generating the PWM output signal on the VS pin.

This Pulse Width modulated signal must be filtered, using an external RC network placed as close as possible to the associated pin. This provides an analog voltage proportional to the average charge passed to the external capacitor. Thus for a higher mark/space ratio (High time much greater than Low time) the average output voltage is higher. The external components of the RC network should be selected for the filtering level required for control of the system variable.

Figure 107. Typical PWM Output Filter

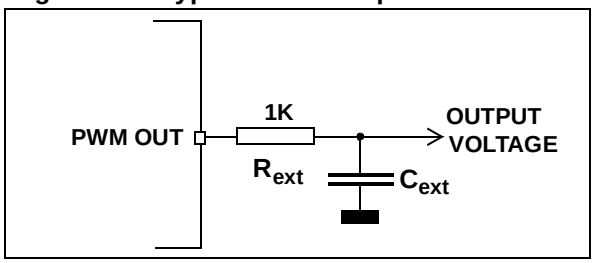
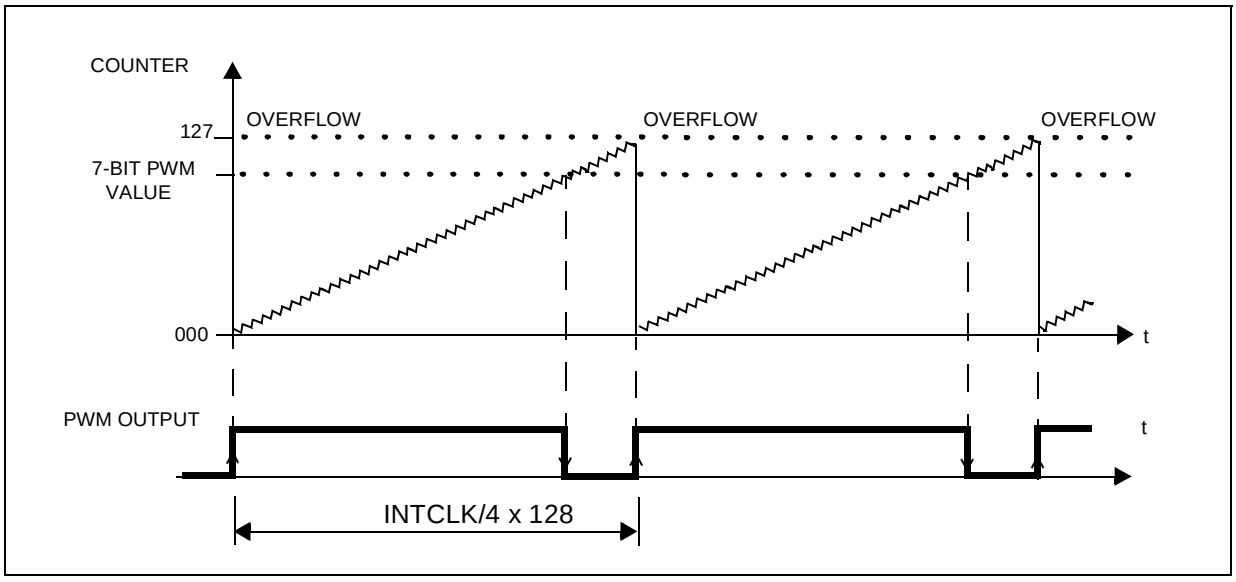
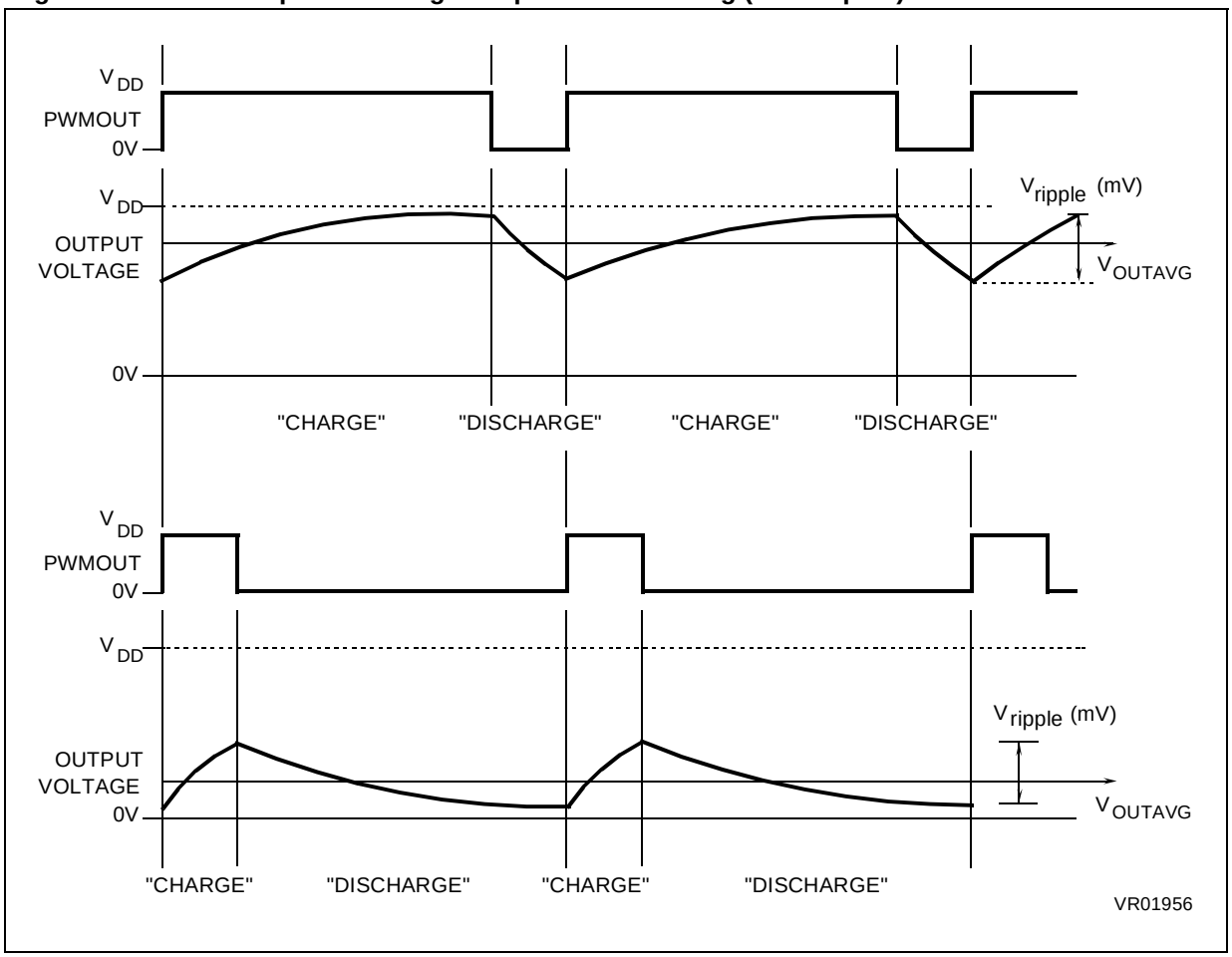


Figure 108. PWM Generation



VOLTAGE SYNTHESIS (Cont'd)

Figure 109. PWM Simplified Voltage Output After Filtering (2 examples)



VOLTAGE SYNTHESIS (Cont'd)

BRM Generation

The BRM bits allow the addition of a pulse to widen a standard PWM pulse for specific PWM cycles. This has the effect of "fine-tuning" the PWM Duty cycle (without modifying the base duty cycle), thus, with the external filtering, providing additional fine voltage steps.

The incremental pulses (with duration of T_{INTCLK}/4) are added to the beginning of the original PWM pulse and thus cause the PWM high time to be extended by this time with a corresponding reduction in the low time. The PWM intervals which are added to are specified in the lower 7 bits of the tuning word and are encoded as shown in the following table.

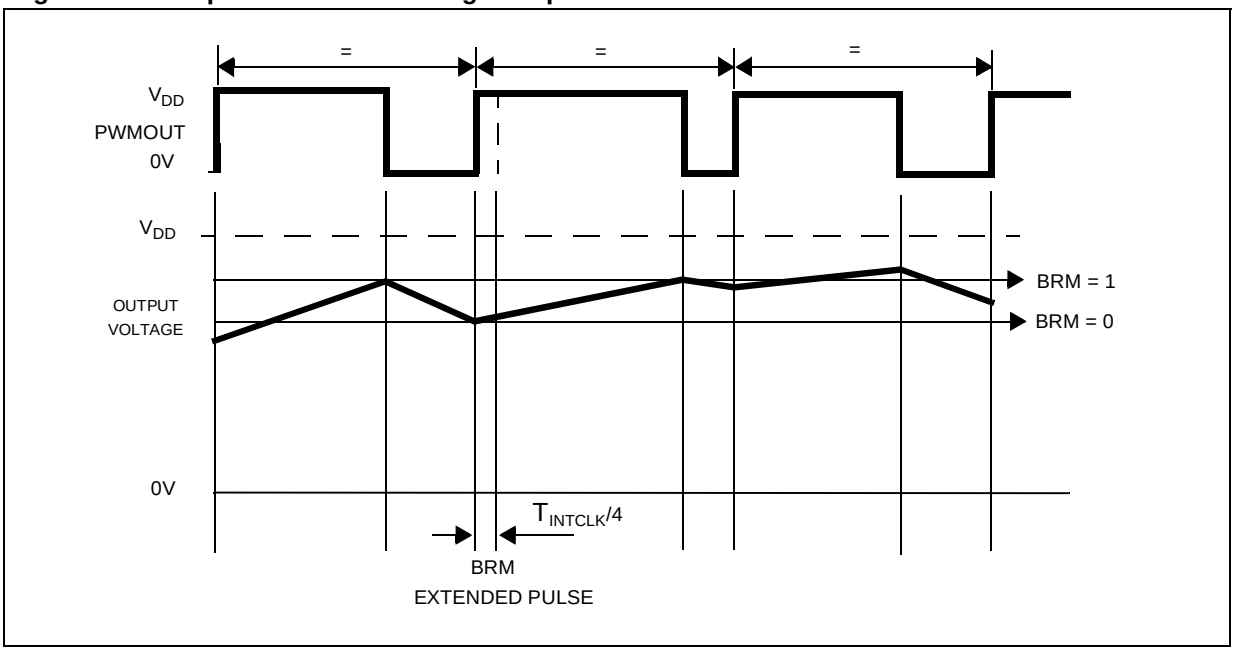
Table 36. 7-Bit BRM Pulse Addition Positions

Fine Tuning	No. of Pulses added at the following Cycles
0000001	64
0000010	32, 96
0000100	16, 48, 80, 112
0001000	8, 24,... 104, 120
0010000	4, 12,... 116, 124
0100000	2, 6,... 122, 126
1000000	1, 3,... 125, 127

The BRM values shown may be combined together to provide a summation of the incremental pulse intervals specified.

The pulse increment corresponds to the PWM resolution.

Figure 110. Simplified Filtered Voltage Output Schematic with BRM added



VOLTAGE SYNTHESIS (Cont'd)

7.12.3 Register Description

VS DATA AND CONTROL REGISTER 1 (VSDR1)
R254 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
VSE	VSWP	VD13	VD12	VD11	VD10	VD9	VD8

Bit 7 = **VSE**: *VS enable bit.*
0: VS Tuning Converter disabled (i.e. the clock is not forwarded to the VS counter and the 2 outputs are set to 1 (idle state)
1: VS Tuning Converter enabled.

Bit 6 = **VSWP**: *VS Output Select*
This bit controls which VS output is enabled to output the VS signal.
0: VSO1 output selected
1: VSO2 output selected

Bit 5:0 = **VD[13:8]** *Tuning word bits.*
These bits are the 6 most significant bits of the Tuning word forming the PWM selection. The VD13 bit is the MSB.

VS DATA AND CONTROL REGISTER 2 (VSDR2)
R255 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
VD7	VD6	VD5	VD4	VD3	VD2	VD1	VD0

Bit 7:0 = **VD[7:0]** *Tuning word bits.*
These bits are the 8 least significant data bits of the VS Tuning word. All bits are accessible. Bits VD6 - VD0 form the BRM pulse selection. VD7 is the LSB of the 7 bits forming the PWM selection.

7.13 PWM GENERATOR

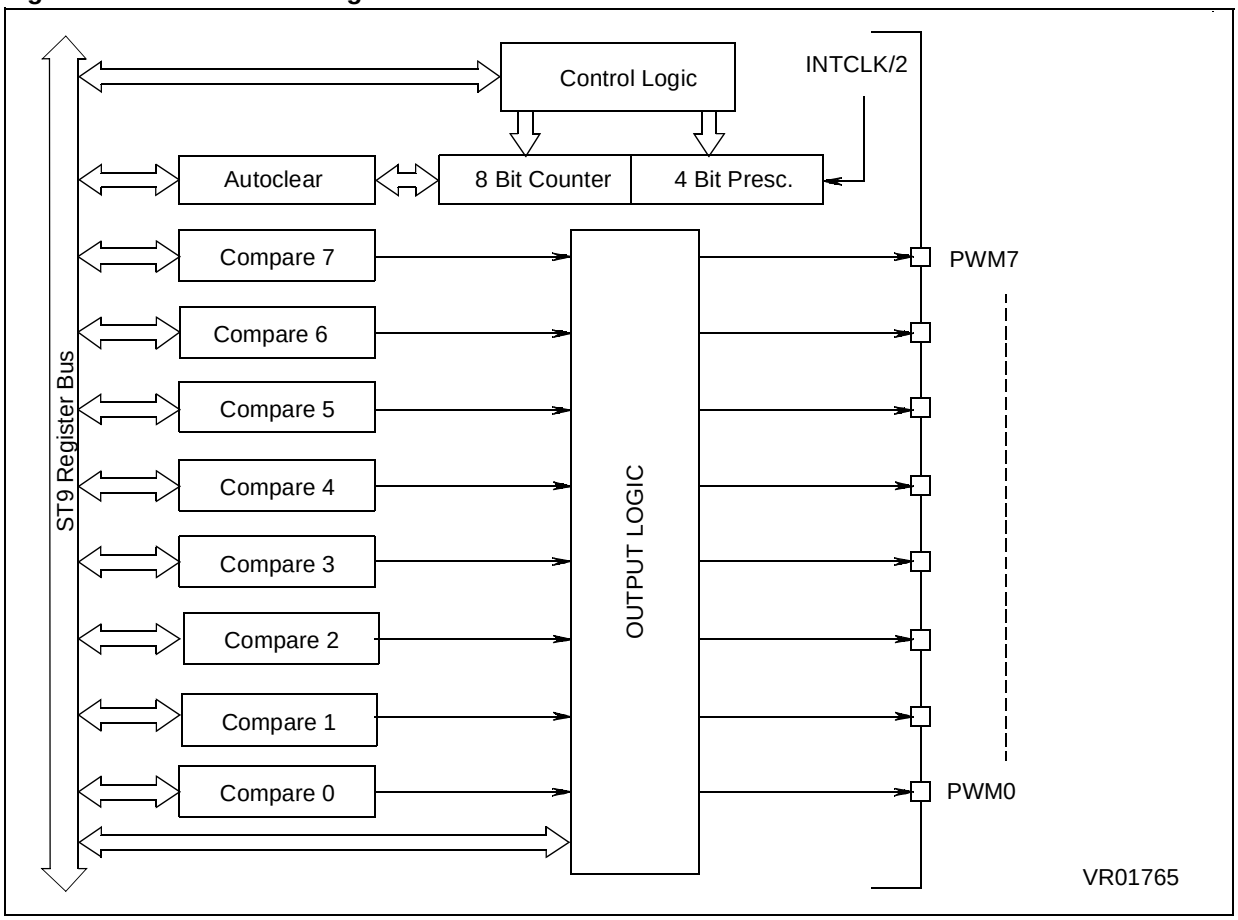
7.13.1 Introduction

The PWM (Pulse Width Modulated) signal generator allows the digital generation of up to 8 analog outputs when used with an external filtering network.

The unit is based around an 8-bit counter which is driven by a programmable 4-bit prescaler, with an input clock signal equal to the internal clock INTCLK divided by 2. For example, with a 12 MHz Internal clock, using the full 8-bit resolution, a frequency range from 1465 Hz up to 23437 Hz can be achieved.
Higher frequencies, with lower resolution, can be achieved by using the autoclear register. As an example, with a 12 MHz Internal clock, a maximum PWM repetition rate of 93750 Hz can be reached with 6-bit resolution.

Note: The number of output pins is device dependant. Refer to the device pinout description.

Figure 111. PWM Block Diagram.



PWM GENERATOR (Cont'd)

Up to 8 PWM outputs can be selected as Alternate Functions of an I/O port. Each output bit is independently controlled by a separate Compare Register. When the value programmed into the Compare Register and the counter value are equal, the corresponding output bit is set. The output bit is reset by a counter clear (by overflow or autoclear), generating the variable PWM signal.

Each output bit can also be complemented or disabled under software control.

7.13.2 Register Mapping

The ST9 can have one or two PWM Generators. Each has 13 registers mapped in page 59 (PWM0) or page 58 (PWM1). In the register description on the following pages, the register page refers to PWM0 only.

Register Address	Register	Function
R240	CM0	Ch. 0 Compare Register
R241	CM1	Ch. 1 Compare Register
R242	CM2	Ch. 2 Compare Register
R243	CM3	Ch. 3 Compare Register
R244	CM4	Ch. 4 Compare Register
R245	CM5	Ch. 5 Compare Register
R246	CM6	Ch. 6 Compare Register
R247	CM7	Ch. 7 Compare Register
R248	ACR	Autoclear Register
R249	CRR	Counter Read Register
R250	PCTLR	Prescaler/ Reload Reg.
R251	OCPLR	Output Complement Reg.
R252	OER	Output Enable Register
R253- R255	—	Reserved

Figure 112. PWM Action When Compare Register = 0 (no complement)

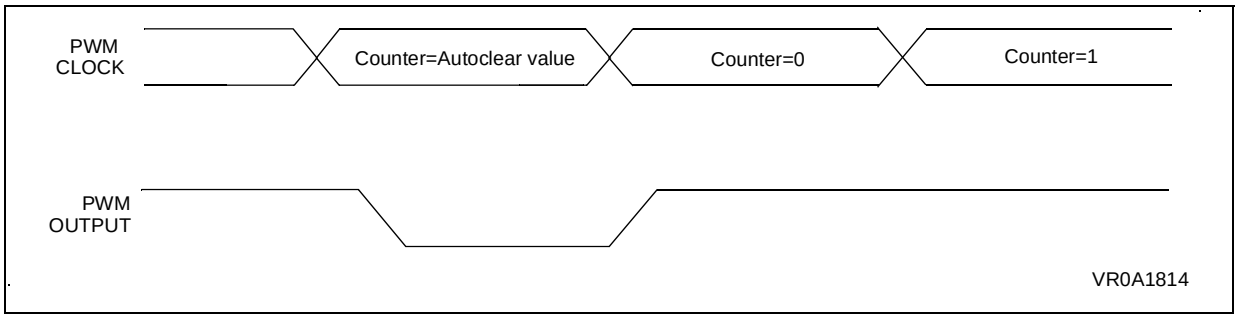
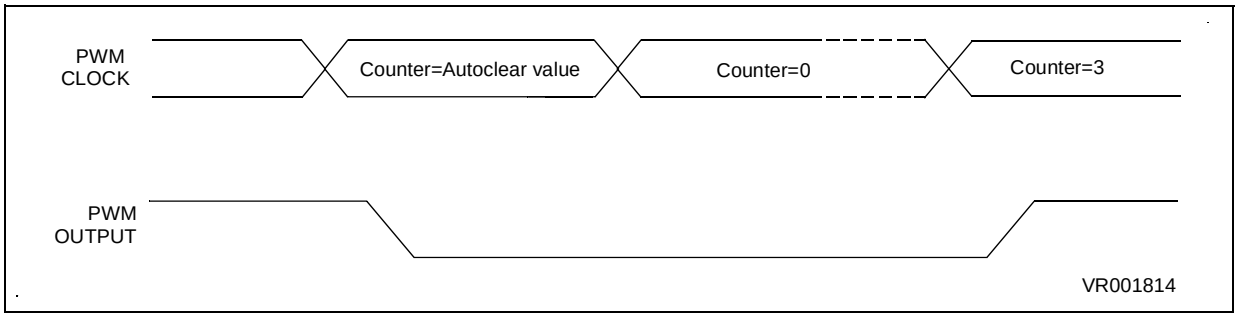


Figure 113. PWM Action When Compare Register = 3 (no complement)



PWM GENERATOR (Cont'd)

7.13.2.1 Register Description

COMPARE REGISTER 0 (CM0)
R240 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM0.7	CM0.6	CM0.5	CM0.4	CM0.3	CM0.2	CM0.1	CM0.0

This is the compare register controlling PWM output 0. When the programmed content is equal to the counter content, a SET operation is performed on PWM output 0 (if the output has not been complemented or disabled).

Bit 7:0 = **CM0.[7:0]**: PWM Compare value Channel 0.

COMPARE REGISTER 1 (CM1)
R241 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM1.7	CM1.6	CM1.5	CM1.4	CM1.3	CM1.2	CM1.1	CM1.0

This is the compare register controlling PWM output 1.

COMPARE REGISTER 2 (CM2)
R242 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM2.7	CM2.6	CM2.5	CM2.4	CM2.3	CM2.2	CM2.1	CM2.0

This is the compare register controlling PWM output 2.

COMPARE REGISTER 3 (CM3)
R243 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM3.7	CM3.6	CM3.5	CM3.4	CM3.3	CM3.2	CM3.1	CM3.0

This is the compare register controlling PWM output 3.

COMPARE REGISTER 4 (CM4)
R244 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM4.7	CM4.6	CM4.5	CM4.4	CM4.3	CM4.2	CM4.1	CM4.0

This is the compare register controlling PWM output 4.

COMPARE REGISTER 5 (CM5)
R245 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM5.7	CM5.6	CM5.5	CM5.4	CM5.3	CM5.2	CM5.1	CM5.0

This is the compare register controlling PWM output 5.

COMPARE REGISTER 6 (CM6)
R246 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM6.7	CM6.6	CM6.5	CM6.4	CM6.3	CM6.2	CM6.1	CM6.0

This is the compare register controlling PWM output 6.

COMPARE REGISTER 7 (CM7)
R247 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CM7.7	CM7.6	CM7.5	CM7.4	CM7.3	CM7.2	CM7.1	CM7.0

This is the compare register controlling PWM output 7.

PWM GENERATOR (Cont'd)

AUTOCLEAR REGISTER (ACR)
R248 - Read/Write
Register Page: 59
Reset Value: 1111 1111 (FFh)

7							0
AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This register behaves exactly as a 9th compare Register, but its effect is to clear the CRR counter register, so causing the desired PWM repetition rate.

The reset condition generates the free running mode. So, FFh means count by 256.

Bit 7:0 = **AC[7:0]**: *Autoclear Count Value.*

When 00 is written to the Compare Register, if the ACR register = FFh, the PWM output bit is always set except for the last clock count (255 set and 1 reset; the converse when the output is complemented). If the ACR content is less than FFh, the PWM output bit is set for a number of clock counts equal to that content (see Figure 2).

Writing the Compare register constant equal to the ACR register value causes the output bit to be always reset (or set if complemented).

Example: If 03h is written to the Compare Register, the output bit is reset when the CRR counter reaches the ACR register value and set when it reaches the Compare register value (after 4 clock counts, see Figure 112.). The action will be reversed if the output is complemented. The PWM mark/space ratio will remain constant until changed by software writing a new value in the ACR register.

COUNTER REGISTER (CRR)
R249 - Read Only
Register Page: 59
Reset Value: 0000 0000 (00h)

7							0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

This read-only register returns the current counter value when read.

The 8 bit Counter is initialized to 00h at reset, and is a free running UP counter.

Bit 7:0 = **CR[7:0]**: *Current Counter Value.*

PRESCALER AND CONTROL REGISTER (PCTL)
R250 - Read/Write
Register Page: 59
Reset Value: 0000 1100 (0Ch)

7							0
PR3	PR2	PR1	PR0	1	1	CLR	CE

Bit 7:4 = **PR[3:0]** *PWM Prescaler value*.
These bits hold the Prescaler preset value. This is reloaded into the 4-bit prescaler whenever the prescaler (DOWN Counter) reaches the value 0, so determining the 8-bit Counter count frequency. The value 0 corresponds to the maximum counter frequency which is INTCLK/2. The value Fh corresponds to the maximum frequency divided by 16 (INTCLK/32).

The reset condition initializes the Prescaler to the Maximum Counter frequency.

PR[3:0]	Divider Factor	Frequency
0	1	INTCLK/2 (Max.)
1	2	INTCLK/4
2	3	INTCLK/6
..	..	..
Fh	16	INTCLK/32 (Min.)

Bit 3:2 = Reserved. Forced by hardware to "1"

Bit 1 = **CLR**: *Counter Clear.*
This bit when set, allows both to clear the counter, and to reload the prescaler. The effect is also to clear the PWM output. It returns "0" if read.

Bit 0 = **CE**: *Counter Enable.*
This bit enables the counter and the prescaler when set to "1". It stops both when reset without affecting their current value, allowing the count to be suspended and then restarted by software "on fly".

PWM GENERATOR (Cont'd)

OUTPUT COMPLEMENT REGISTER (OCPL)
R251- Read/Write
Register Page 59
Reset Value: 0000 0000 (00h)

7							0
OC-PL.7	OC-PL.6	OC-PL.5	OC-PL.4	OC-PL.3	OC-PL.2	OC-PL.1	OC-PL.0

This register allows the PWM output level to be complemented on an individual bit basis.

In default mode (reset configuration), each comparison true between a Compare register and the counter has the effect of setting the corresponding output.

At counter clear (either by autoclear comparison true, software clear or overflow when in free running mode), all the outputs are cleared.

By setting each individual bit (OCPL.x) in this register, the logic value of the corresponding output will be inverted (i.e. reset on comparison true and set on counter clear).

Example: When set to "1", the OCPL.1 bit complements the PWM output 1.

Bit 7 = **OCPL.7**: *Complement PWM Output 7.*

Bit 6 = **OCPL.6**: *Complement PWM Output 6.*

Bit 5 = **OCPL.5**: *Complement PWM Output 5.*

Bit 4 = **OCPL.4**: *Complement PWM Output 4.*

Bit 3 = **OCPL.3**: *Complement PWM Output 3.*

Bit 2 = **OCPL.2**: *Complement PWM Output 2.*

Bit 1 = **OCPL.1**: *Complement PWM Output 1.*

Bit 0 = **OCPL.0**: *Complement PWM Output 0.*

OUTPUT ENABLE REGISTER (OER)
R252 - Read/Write
Register Page: 59
Reset Value: 0000 0000 (00h)

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

These bits are set and cleared by software.
0: Force the corresponding PWM output to logic level 1. This allows the port pins to be used for normal I/O functions or other alternate functions (if available).
1: Enable the corresponding PWM output.

Example: Writing 03h into the OE Register will enable only PWM outputs 0 and 1, while outputs 2, 3, 4, 5, 6 and 7 will be forced to logic level "1".

Bit 7 = **OE.7**: *Output Enable PWM Output 7.*

Bit 6 = **OE.6**: *Output Enable PWM Output 6.*

Bit 5 = **OE.5**: *Output Enable PWM Output 5.*

Bit 4 = **OE.4**: *Output Enable PWM Output 4.*

Bit 3 = **OE.3**: *Output Enable PWM Output 3.*

Bit 2 = **OE.2**: *Output Enable PWM Output 2.*

Bit 1 = **OE.1**: *Output Enable PWM Output 1.*

Bit 0 = **OE.0**: *Output Enable PWM Output 0.*

8 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	V_{SS} - 0.3 to V_{SS} + 7.0	V
V_{SSA}	Analog Ground	V_{SS} - 0.3 to V_{SS} + 0.3	V
V_{DDA}	Analog Supply Voltage	V_{DD} -0.3 to V_{DD} +0.3	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} +0.3	V
V_{AI}	Analog Input Voltage (A/D Converter)	V_{SS} - 0.3 to V_{DD} +0.3 V_{SSA} - 0.3 to V_{DDA} +0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	- 55 to + 150	°C
I_{INJ}	Pin Injected Current Maximum Accumulated Pin Injected Current In Device	- 5 to + 5 - 50 to +5 0	mA

Note: Stress above those listed as “Absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Supply Voltage	4.5	5.5	V
V_{DDA}	Analog Supply Voltage (PLL)	4.5	5.5	V

Note: In order for the data slicer to operate correctly, the crystal frequency must be either 4 or 8 MHz. Please refer to Section 4, RESET AND CLOCK CONTROL UNIT (RCCU).

DC ELECTRICAL CHARACTERISTICS

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Value		Unit
			Min.	Max.	
V_{IHCK}	Clock In high level	External clock	0.7 V_{DD}		V
V_{ILCK}	Clock in low level	External clock		0.3 V_{DD}	V
V_{IH}	Input high level	TTL	2.0		V
V_{IL}	Input low level	TTL		0.8	V
V_{IH}	Input high level	CMOS	0.7 V_{DD}		V
V_{IL}	Input low level	CMOS		0.3 V_{DD}	V
V_{IHRS}	Reset in high level		0.7 V_{DD}		V
V_{ILRS}	Reset in low level			0.3 V_{DD}	V
V_{HYRS}	Reset in hysteresis		0.3		V
V_{IHY}	P2.(1:0) input hysteresis		0.9		V
V_{IHVH}	HSYNC/VSYNC input high level		0.7 V_{DD}		V
V_{ILVH}	HSYNC/VSYNC input low level			0.3 V_{DD}	V
V_{HYHV}	HSYNC/VSYNC input hysteresis		0.5		V
V_{OH}	Output high level	Push-pull I_{LOAD} =-0.8 mA	V_{DD}-0.8		V
V_{OL}	Output low level	Push-pull I_{LOAD} =+1.6 mA		0.4	V
I_{WPU}	Weak pull-up current	Bidirectional weak pull-up $V_{IN} = V_{SS}$	-50	-420	µA
I_{LKIO}	I/O pin input leakage current	$0<V_{IN}<V_{DD}$	-1	+1	µA
I_{LKRS}	Reset pin input	$0<V_{IN}<V_{DD}$	-1	+1	µA
I_{LKAD}	A/D pin input leakage current	alternate funct. op. drain	-1	+1	µA
I_{LKOS}	OSCIN pin input leakage current	$0<V_{IN}<V_{DD}$	-1	+1	µA

AC ELECTRICAL CHARACTERISTICS

PIN CAPACITANCE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Conditions	Value		Unit
			min	max	
C_{IO}	Pin Capacitance Digital Input/Output			10	pF

CURRENT CONSUMPTION

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Conditions	Value			Unit
			min	typ.	max	
I_{DD1}	Run Mode Current	Notes 1,2; all ON		70	100	mA
I_{DDA1}	Run Mode Analog Current (pin V_{DDA})	Timing Controller ON		35	50	mA
I_{DD2}	HALT Mode Digital Current	Notes 1,4		10	100	µA
I_{DDA2}	HALT Mode Analog Current (pin V_{DDA})	Notes 1,4		10	100	µA

Notes:

1. Port 0 is configured in push-pull output mode (output is high). Ports 2, 3, 4 and 5 are configured in bi-directional weak pull-up mode resistor. The external CLOCK pin (OSCIN) is driven by a square wave external clock at 8 MHz. The internal clock prescaler is in divide-by-1 mode.

2. The CPU is fed by a 24 MHz frequency issued by the Main Clock Controller. VSYNC is tied to V_{SS}, HSYNC is driven by a 15625 Hz clock. All peripherals working including Display.

3. The CPU is fed by a 24 MHz frequency issued by the Main Clock Controller. VSYNC is tied to V_{SS}, HSYNC is driven by a 15625 Hz clock. The TDSRAM interface and the Slicers are working; the Display controller is not working.

4. VSYNC and HSYNC tied to V_{SS}. External CLOCK pin (OSCIN) is hold low. All peripherals are disabled.

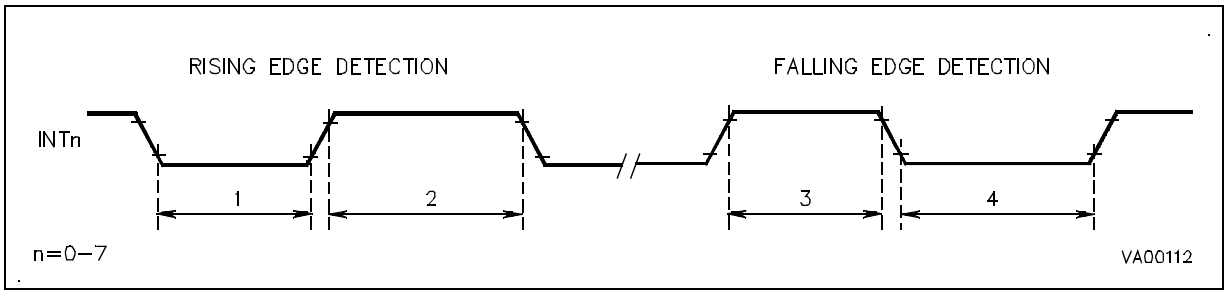
EXTERNAL INTERRUPT TIMING TABLE

(V_{DD} = 5 V ± 10%, T_A = 0°C +70° C, C_{LOAD} = 50 pF, INTCLK = 12 MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	**Value** (Note)			Unit
			OSCIN Divided by 2 Min.	OSCIN Not Divided by 2 Min.	Min.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95	ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95	ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95	ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95	ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.
TpC = OSCIN clock period

EXTERNAL INTERRUPT TIMING



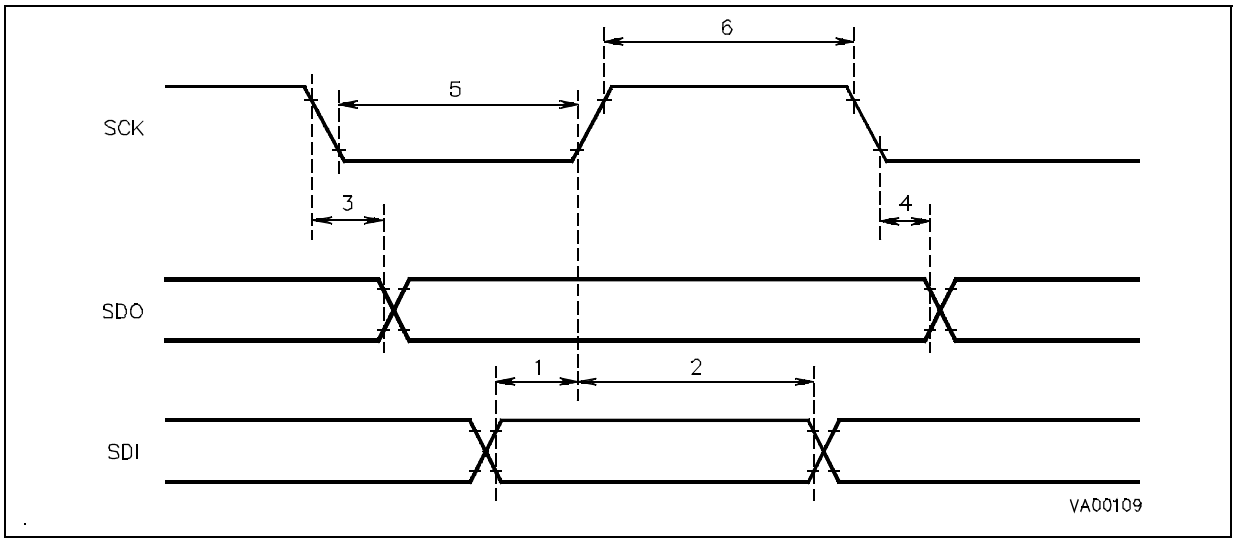
SPI TIMING TABLE

(V_{DD} = 5 V ± 10%, T_A = 0° C +70° C, Cload = 50 pF, INTCLK = 12 MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

SPI TIMING



SKEW CORRECTOR TIMING TABLE

(V_{DD}= 5V +/-10%, T_A= 0 to 70°C, unless otherwise specified)

Symbol	Parameter	Conditions	max Value	Unit
T_{jskw}	Jitter on RGB output	36 MHz Skew corrector clock frequency	5*	ns

(*) The OSD jitter is measured from leading edge to leading edge of a single character row on consecutive TV lines. The value is an envelope of 100 fields

OSD DAC CHARACTERISTICS

(V_{DD}= 5V +/-10%, T_A= 0 to 70°C, unless otherwise specified).

Symbol	Parameter	Conditions	Value			Unit
			min	typical	max	
	Output impedance: FB,R,G,B		300	500	700	Ohm
	Output voltage: FB,R,G,B	C_{LOAD}= 20 pF R_{LOAD} = 100 kΩ				
	code= 111			1.100	1.400	V
	code= 011			0.600	0.800	V
	code= 000			0.200	0.350	V
	FB= 1		V_{DD}-0.8			V
	FB= 0				0.400	V
	Global voltage accuracy				+/-5	%

A/D CONVERTER, EXTERNAL TRIGGER TIMING TABLE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	OSCIN divide by 2;min/max	OSCIN divide by 1; min/max	Value		Unit
				min	max	
T_{low}	Pulse Width			1.5 INTCLK		ns
T_{ext}	Period/fast Mode			78+1 INTCLK		µs
T_{str}	Start Conversion Delay			0.5	1.5	INTCLK

A/D CONVERTER. ANALOG PARAMETERS TABLE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Parameter	Value			Unit	Note
	typ (*)	min	max	(**)	
Analog Input Range		V_{SS}	V_{DD}	V	
Conversion Time Fast/Slow		78/138		INTCLK	(1,2)
Sample Time Fast/Slow		51.5/87.5		INTCLK	(1)
Power-up Time		60		µs	
Resolution	8			bits	
Differential Non Linearity	1.5		2.5	LSBs	(4)
Integral Non Linearity	2		3	LSBs	(4)
Absolute Accuracy	2		3	LSBs	(4)
Input Resistance			1.5	Kohm	(3)
Hold Capacitance			1.92	pF	

Notes:(*)The values are expected at 25 Celsius degrees with V_{DD}= 5V
(**)'LSBs' , as used here, as a value of V_{DD}/256
(1)@ 24 MHz external clock
(2)including Sample time
(3)it must be considered as the on-chip series resistance before the sampling capacitor
(4)DNL ERROR= max {[V(i) -V(i-1)] / LSB-1}INL ERROR= max {[V(i) -V(0)] / LSB-i}
ABSOLUTE ACCURACY= overall max conversion error

I²C BUS TIMING

Symbol	Parameter	Standard I²C		Fast I²C		Unit
		Min	Max	Min	Max	
T_{BUF}	Bus free time between a STOP and START condition	4.7		1.3		ms
$T_{HD:STA}$	Hold time START condition. After this period, the first clock pulse is generated	4.0		0.6		µs
T_{LOW}	LOW period of the SCL clock	4.7		1.3		µs
T_{HIGH}	HIGH period of the SCL clock	4.0		0.6		µs
$T_{SU:STA}$	Set-up time for a repeated START condition	4.7		0.6		µs
$T_{HD:DAT}$	Data hold time	0 (1)		0 (1)	0.9(2)	ns
$T_{SU:DAT}$	Data set-up time	250		100		ns
T_R	Rise time of both SDA and SCL signals		1000	20+0.1Cb	300	ns
TF	Fall time of both SDA and SCL signals		300	20+0.1Cb	300	ns
$T_{SU:STO}$	Set-up time for STOP condition	4.0		0.6		ns
Cb	Capacitive load for each bus line		400		400	pF

1)The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL

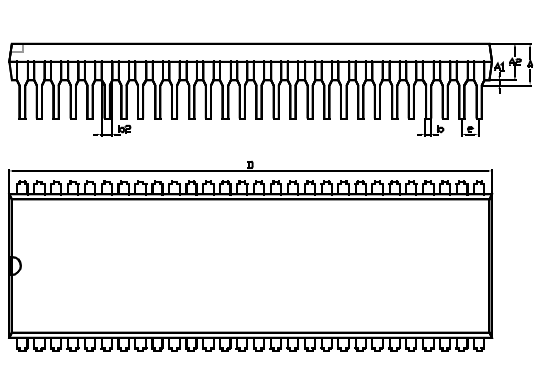
2)The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal

Cb = total capacitance of one bus line in pF

9 GENERAL INFORMATION

9.1 PACKAGE MECHANICAL DATA

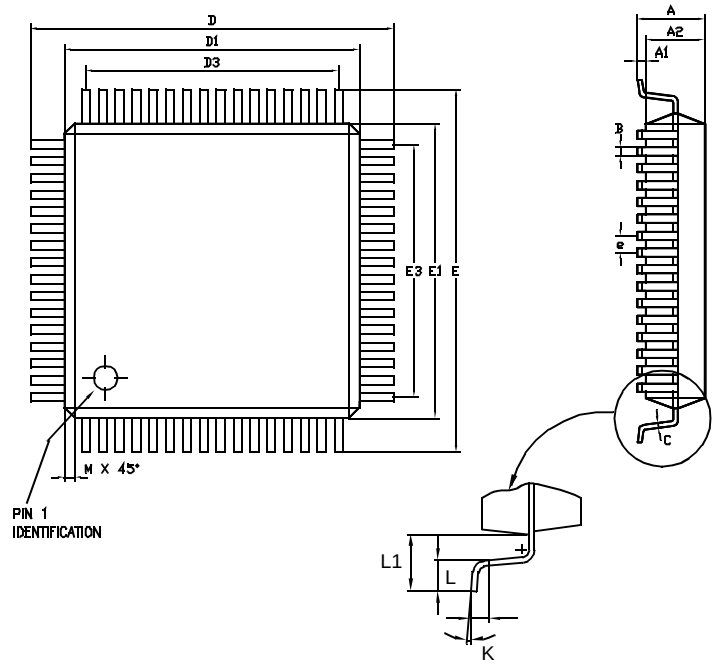
Figure 114. 56-Pin Shrink Plastic Dual In Line Package, 600-mil Width



PDIP56S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			6.35			0.250
A1	0.38			0.015		
A2	3.18		4.95	0.125		0.195
b		0.41			0.016	
b2		0.89			0.035	
C	0.20		0.38	0.008		0.015
D	50.29		53.21	1.980		2.095
E		15.01			0.591	
E1	12.32		14.73	0.485		0.580
e		1.78			0.070	
eA		15.24			0.600	
eB			17.78			0.700
L	2.92		5.08	0.115		0.200
	Number of Pins					
N	56					

Figure 115. 64-Pin Thin Quad Flat Package



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.30	0.37	0.45	0.012	0.015	0.018
C	0.09		0.20	0.004		0.008
D		16.00			0.630	
D1		14.00			0.551	
D3		12.00			0.472	
E		16.00			0.630	
E1		14.00			0.551	
E3		12.00			0.472	
e		0.80			0.031	
K	0°	3.5°	7°			
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	64		ND	16	NE	16

9.2 ORDERING INFORMATION

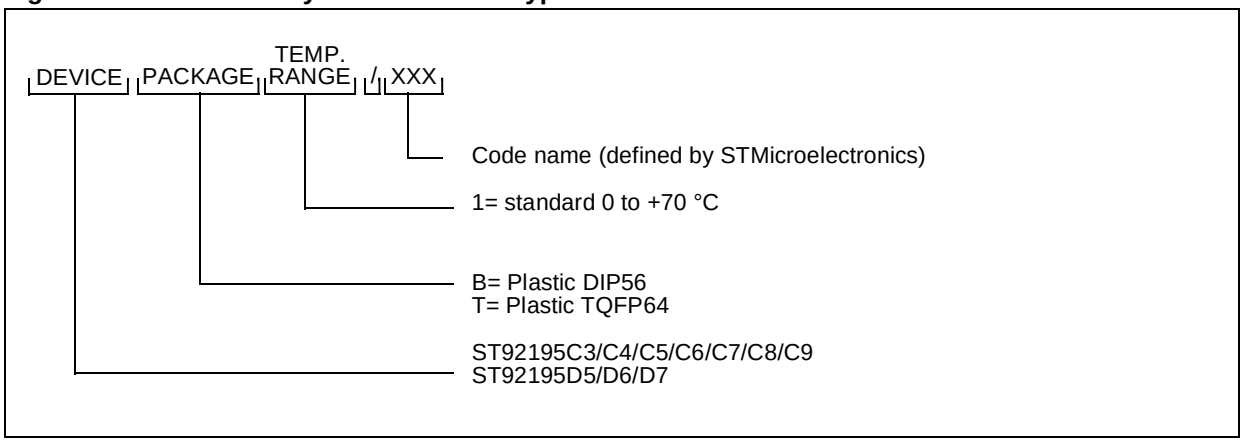
Each device is available for production in a user programmable version (OTP) as well as in factory coded version (ROM). OTP devices are shipped to customer with a default blank content FFh, while ROM factory coded parts contain the code sent by customer. The common EPROM versions for debugging and prototyping features the maximum memory size and peripherals of the family. Care must be taken to only use resources available on the target device.

9.2.1 Transfer Of Customer Code

Customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The OSD files must be sent in 2 files named xxx.OS0 and xxx.OS1.

Figure 116. ROM Factory Coded Device Types



STMicroelectronics OPTION LIST

ST92185 AND ST92195

ST92E195C/D-
ST92T195C/D

64-84K EPROM/OTP HCMOS MCU WITH ON-SCREEN-DISPLAY AND TELETEXT DATA SLICER

Customer: . . . . . .

Address: . . . . . .

. . . . . .

Contact: . . . . . .

Phone No: . . . . . .

Fax No: . . . . . .

STMicroelectronics reference:

Device: [ ] ST92185B2 [ ] ST92185B3

[ ] ST92195B1 [ ] ST92195B3 [ ] ST92195B4
[ ] ST92195B5 [ ] ST92195B6 [ ] ST92195B7

[ ] ST92195C3 [ ] ST92195C4 [ ] ST92195C5
[ ] ST92195C6 [ ] ST92195C7 [ ] ST92195C8
[ ] ST92195C9

[ ] ST92195D5 [ ] ST92195D6 [ ] ST92195D7

Package : [ ] PSDIP42* [ ] PSDIP56 [ ] TQFP64
(*ST92185 only)

Temperature Range : 0° to 70° C

Software Development: [ ] STMicroelectronics
[ ] Customer
[ ] External laboratory

Special Marking: [ ] No [ ] Yes "_____"
SDIP42 "_____"
PSDIP56 "_____"
TQFP64 "_____"

For marking, one line is possible with maximum 14 characters (SDIP42/56) and 10 characters (TQFP64). Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Please consult your local ST Microelectronics sales office for other marking details if required.

Notes :

OSD Code : [ ] OSD Files Filename [..... .OS0] Filename [..... .OS1]
: [ ] OSD File Filename [..... .OSD] (for ST92185B and ST92195B)

Quantity forecast : [.....] k units per year

For a period of : [.....] years

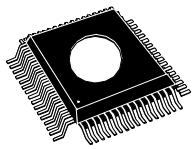
Preferred Production start dates : [../../..] (YY/MM/DD)

Date:

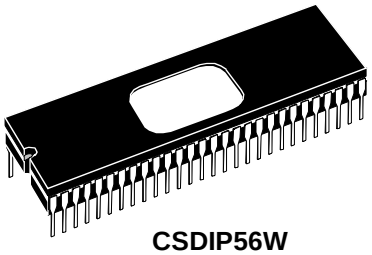
Customer Signature :

- Register File based 8/16 bit Core Architecture with RUN, WFI, SLOW and HALT modes
- 0°C to +70°C operating temperature range
- Up to 24 MHz. operation @ 5V±10%
- Min instruction cycle time: 165ns at 24 MHz.
- 64 or 84 Kbytes EPROM or OTP
- 256 bytes RAM of Register file (accumulators or index registers)
- 512 bytes of on-chip static RAM
- 8 Kbytes of TDSRAM (Teletext and Display Storage RAM)
- 28 fully programmable I/O pins
- Serial Peripheral Interface
- Flexible Clock controller for OSD, Data Slicer and Core clocks running from a single low frequency external crystal.
- Enhanced display controller with 26 rows of 40/80 characters
  - 2 sets of 512 characters, programmable on some devices
  - Serial or parallel attributes
  - 10x10 dot matrix, definable by user
  - 4/3 and 16/9 supported in 50/60Hz and 100/120 Hz mode
  - Rounding, fringe, double width, double height, scrolling, cursor, full background color, half-intensity color, translucency and half-tone modes

- Teletext unit, including Data Slicer, Acquisition Unit and 8 Kbytes RAM for data storage
- VPS and Wide Screen Signalling slicer (on some devices)
- Integrated Sync Extractor and Sync Controller
- 14-bit Voltage Synthesis for tuning reference voltage
- Up to 6 external interrupts plus one Non-Maskable Interrupt
- 8 x 8-bit programmable PWM outputs with 5V open-drain or push-pull capability
- 16-bit watchdog timer with 8-bit prescaler
- 2 16-bit standard timer(s) with 8-bit prescaler



CQFP64W



CSDIP56W

See end of Datasheet for ordering information

- I²C Master/Slave
- 4-channel A/D converter; 5-bit guaranteed
- Rich instruction set and 14 addressing modes
- Versatile development tools, including Assembler, Linker, C-compiler, Archiver, Source Level Debugger and hardware emulators with Real-Time Operating System available from third parties

Device Summary

Device	Program Memory	RAM	TDS RAM	Program-mable OSD font
ST92E195C8	84K EPROM	512	8K	No
ST92T195C8	84K OTP			
ST92E195D7	64K EPROM			Yes
ST92T195D7	64K OTP			

1 GENERAL DESCRIPTION

The ST92E195C/D and ST92T195C/D microcontrollers are the EPROM/OTP versions of the ST92195C/D ROM devices and are suitable for product prototyping and low volume production. Their performance derives from the use of a flexible 256-register programming model for ultra-fast context switching and real-time event response. The intelligent on-chip peripherals offload the ST9 core from I/O and data management processing tasks allowing critical application tasks to get the maximum use of core resources. The ST92195E195C/D and ST92195T195C/D MCUs support low power consumption and low voltage operation for power-efficient and low-cost embedded systems.

1.0.1 ST9+ Core

The advanced Core consists of the Central Processing Unit (CPU), the Register File and the Interrupt controller.

The general-purpose registers can be used as accumulators, index registers, or address pointers. Adjacent register pairs make up 16-bit registers for addressing or 16-bit processing. Although the ST9 has an 8-bit ALU, the chip handles 16-bit operations, including arithmetic, loads/stores, and memory/register and memory/memory exchanges.

Two basic addressable spaces are available: the Memory space and the Register File, which includes the control and status registers of the on-chip peripherals.

1.0.2 Power Saving Modes

To optimize performance versus power consumption, a range of operating modes can be dynamically selected.

Run Mode. This is the full speed execution mode with CPU and peripherals running at the maximum clock speed delivered by the Phase Locked Loop (PLL) of the Clock Control Unit (CCU).

Wait For Interrupt Mode. The Wait For Interrupt (WFI) instruction suspends program execution until an interrupt request is acknowledged. During WFI, the CPU clock is halted while the peripheral and interrupt controller keep running at a frequency programmable via the CCU. In this mode, the power consumption of the device can be reduced by more than 95% (Low power WFI).

Halt Mode. When executing the HALT instruction, and if the Watchdog is not enabled, the CPU and its peripherals stop operating and the status of the machine remains frozen (the clock is also stopped). A reset is necessary to exit from Halt mode.

1.0.3 I/O Ports

Up to 28 I/O lines are dedicated to digital Input/ Output. These lines are grouped into up to five I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer and output, analog inputs, external interrupts and serial or parallel I/O.

1.0.4 TV Peripherals

A set of on-chip peripherals form a complete system for TV set and VCR applications:

– Voltage Synthesis

– VPS/WSS Slicer

– Teletext Slicer

– Teletext Display RAM

– OSD

1.0.5 On Screen Display

The human interface is provided by the On Screen Display module, this can produce up to 26 lines of up to 80 characters from a ROM of two 512-character sets. The character resolution is 10x10 dot. Four character sizes are supported. Serial attributes allow the user to select foreground and background colors, character size and fringe background. Parallel attributes can be used to select additional foreground and background colors and underline on a character by character basis.

Note: ST92E195D7 and ST92T195D7 have programmable OSD fonts.

1.0.6 Teletext and Display Storage RAM

The internal Teletext and Display storage RAM can be used to store Teletext pages as well as Display parameters.

INTRODUCTION (Cont'd)

1.0.7 Teletext, VPS and WSS Data Slicers

The three on-board data slicers using a single external crystal are used to extract the Teletext, VPS and WSS information from the video signal. Hardware Hamming decoding is provided.

1.0.8 Voltage Synthesis Tuning Control

14-bit Voltage Synthesis using the PWM (Pulse Width Modulation)/BRM (Bit Rate Modulation) technique can be used to generate tuning voltages for TV set applications. The tuning voltage is output on one of two separate output pins.

1.0.9 PWM Output

Control of TV settings can be made with up to eight 8-bit PWM outputs, with a maximum frequency of 23,437Hz at 8-bit resolution (INTCLK = 12 MHz). Low resolutions with higher frequency operation can be programmed.

1.0.10 Serial Peripheral Interface (SPI)

The SPI bus is used to communicate with external devices via the SPI, or I²C bus communication standards. The SPI uses a single data line for data input and output. A second line is used for a synchronous clock signal.

1.0.11 Standard Timer (STIM)

The ST92E195C/D and ST92T195C/D have two Standard Timers that include a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes.

1.0.12 I²C Bus Interface

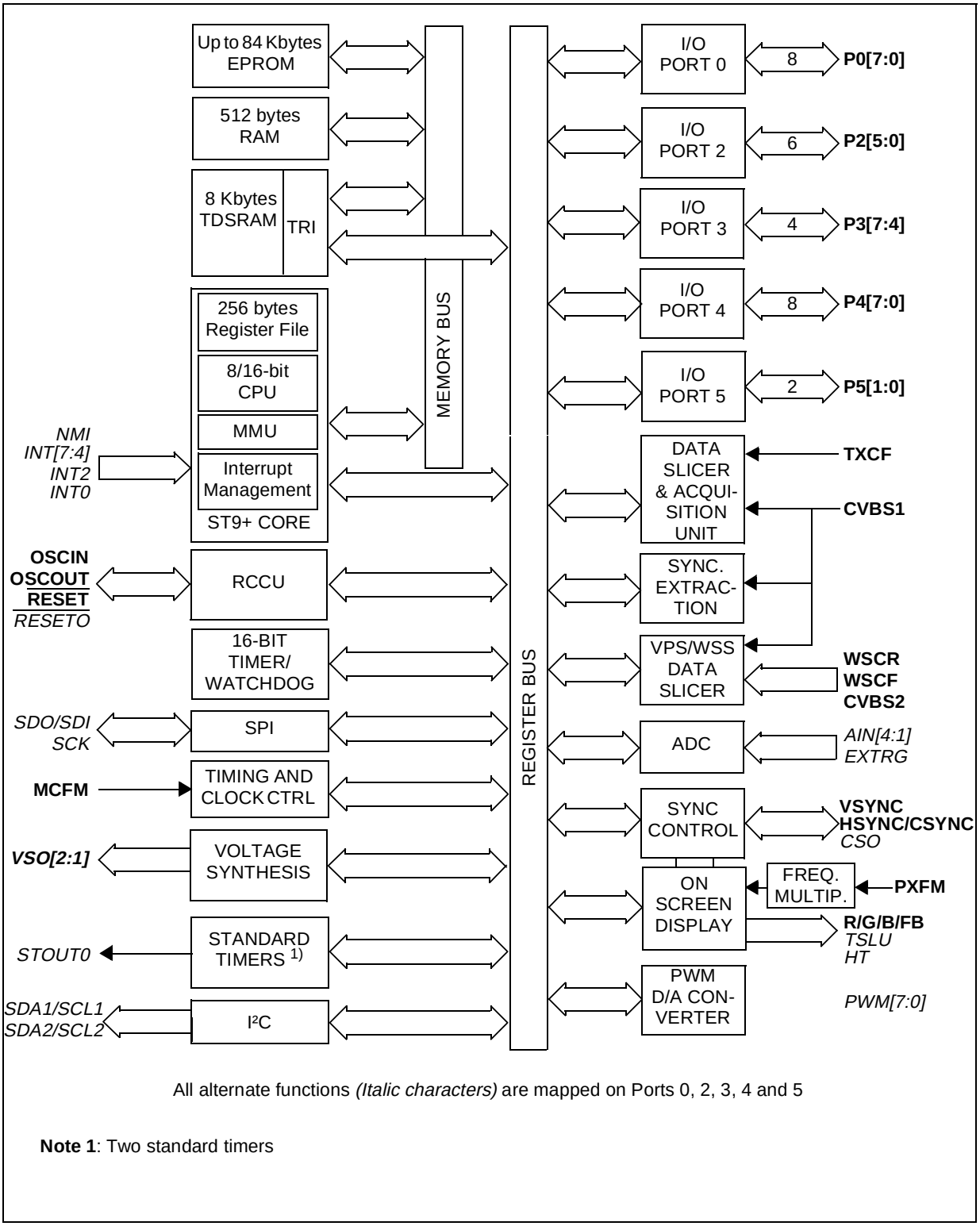
The I²C bus is a synchronous serial bus for connecting multiple devices using a data line and a clock line. Multimaster and slave modes are supported. Up to two channels are supported. The I²C interface supports 7-bit addressing. It supports speeds of up to 800 KHz. Bus events (Bus busy, slave address recognised) and error conditions are automatically flagged in peripheral registers and interrupts are optionally generated.

1.0.13 Analog/Digital Converter (ADC)

In addition there is a 4-channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

INTRODUCTION (Cont'd)

Figure 117. ST92E195C/D-ST92T195C/D Block Diagram

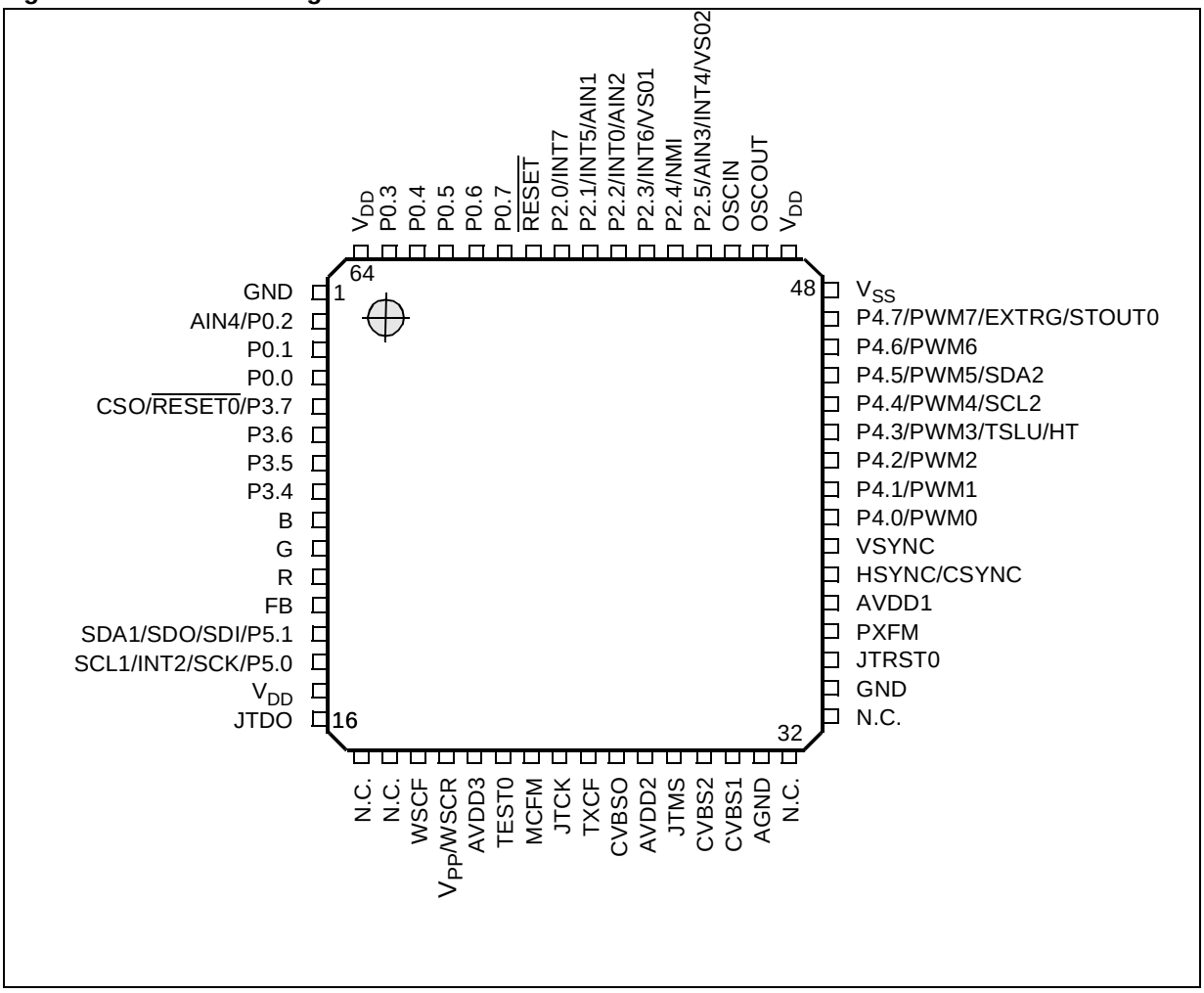


All alternate functions *(Italic characters)* are mapped on Ports 0, 2, 3, 4 and 5

Note 1: Two standard timers

1.1 PIN DESCRIPTION

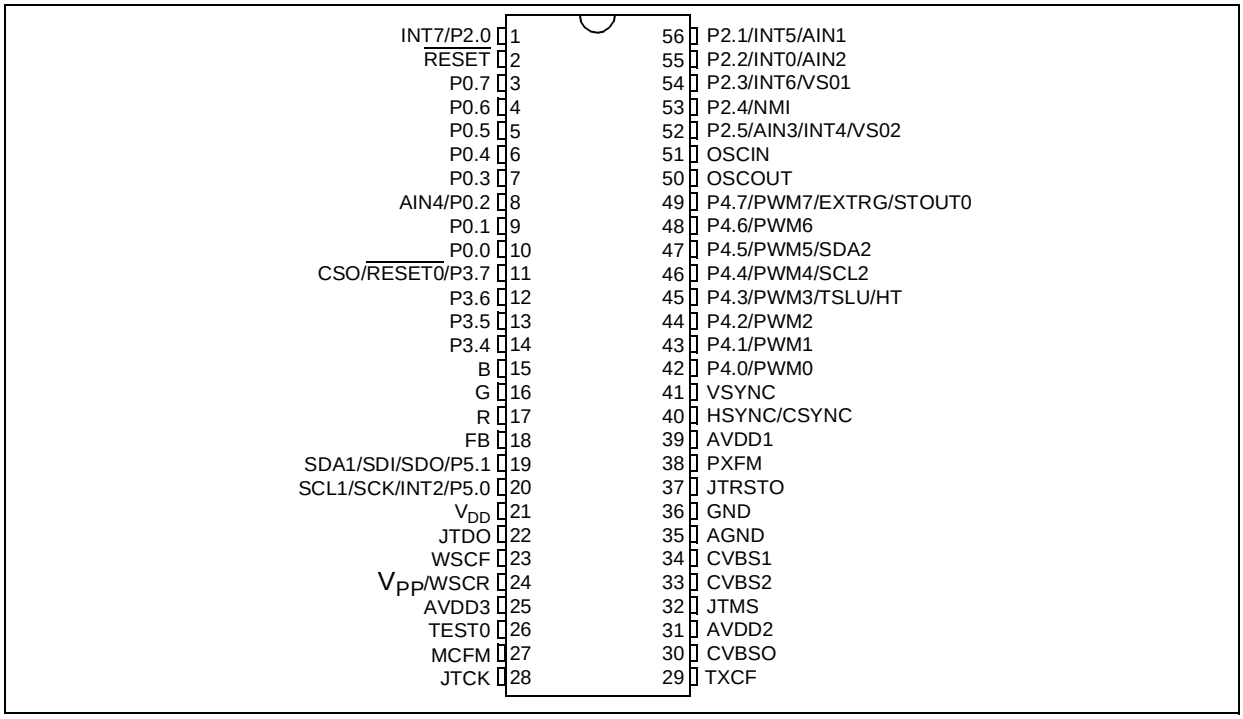
Figure 118. 64-Pin Package Pin-Out



N.C. = Not connected

PIN DESCRIPTION (Cont'd)

Figure 119. 56-Pin Package Pin-Out



RESET *Reset* (input, active low). The ST9+ is initialised by the Reset signal. With the deactivation of RESET, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

R/G/B *Red/Green/Blue*. Video color analog DAC outputs.

FB *Fast Blanking*. Video analog DAC output.

V_{DD} Main power supply voltage (5V±10%, digital)

WSCF, WSCR Analog pins for the VPS/WSS slicer . These pins must be tied to ground or not connected.

V_{PP}: On EPROM/OTP devices, the WSCR pin is replaced by V_{PP} which is the programming voltage pin. V_{PP} should be tied to GND in user mode.

MCFM Analog pin for the display pixel frequency multiplier.

OSCIN, OSCOUT *Oscillator* (input and output). These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

VSYNC *Vertical Sync*. Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC/CSYNC *Horizontal/Composite sync*. Horizontal or composite video synchronisation input to OSD. Positive or negative polarity.

PXFM Analog pin for the Display Pixel Frequency Multiplier

AVDD3 *Analog V_{DD} of PLL.* This pin must be tied to V_{DD} externally.

GND Digital circuit ground.

AGND Analog circuit ground (must be tied externally to digital GND).

CVBS1 Composite video input signal for the Teletext slicer and sync extraction.

CVBS2 Composite video input signal for the VPS/WSS slicer. Pin AC coupled.

AVDD1, AVDD2 Analog power supplies (must be tied externally to AVDD3).

TXCF Analog pin for the Teletext slicer line PLL.

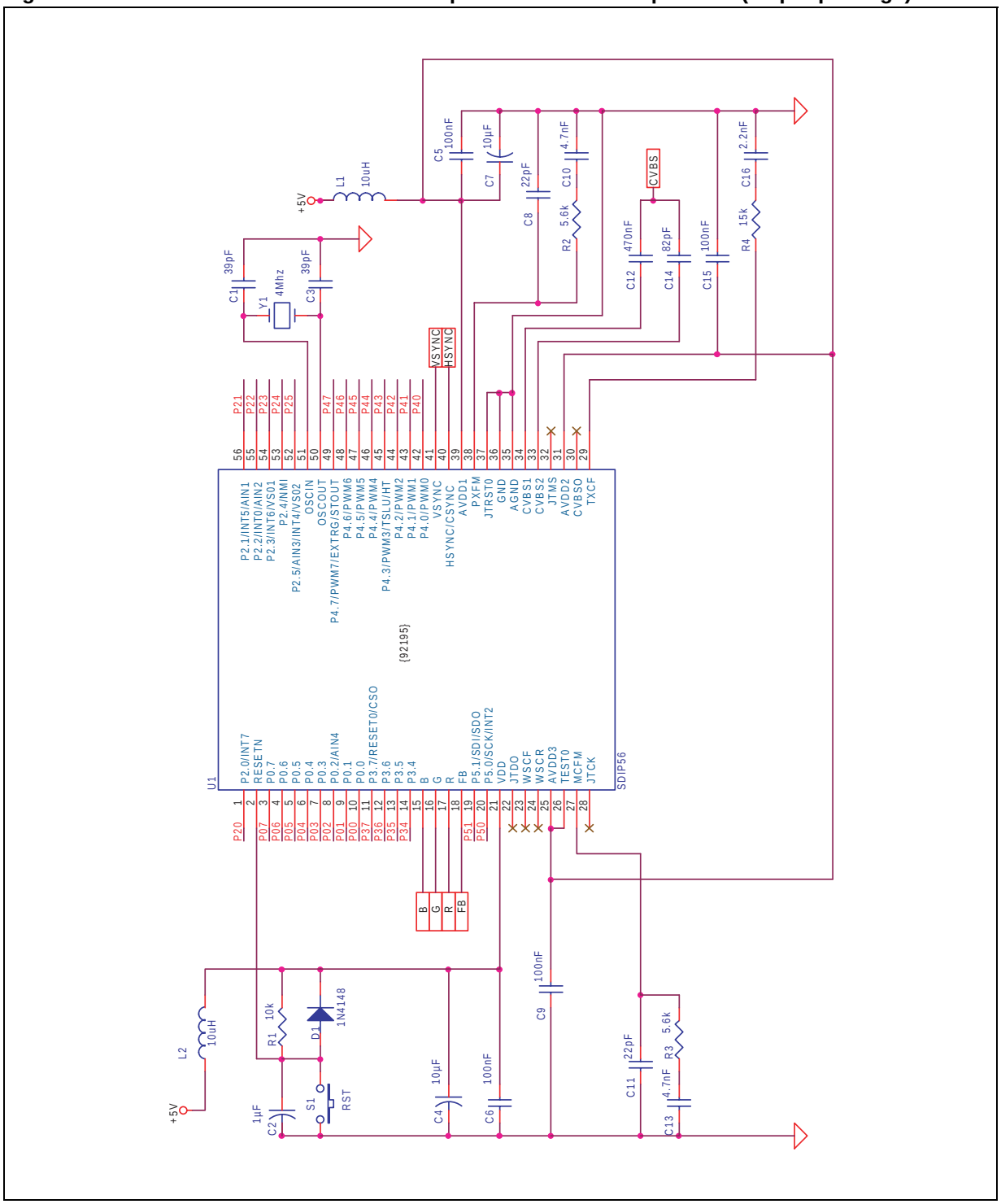
CVBSO, JTDO, JTCK Test pins: leave floating.

TEST0 Test pins: must be tied to AVDD2.

JTRST0 Test pin: must be tied to GND.

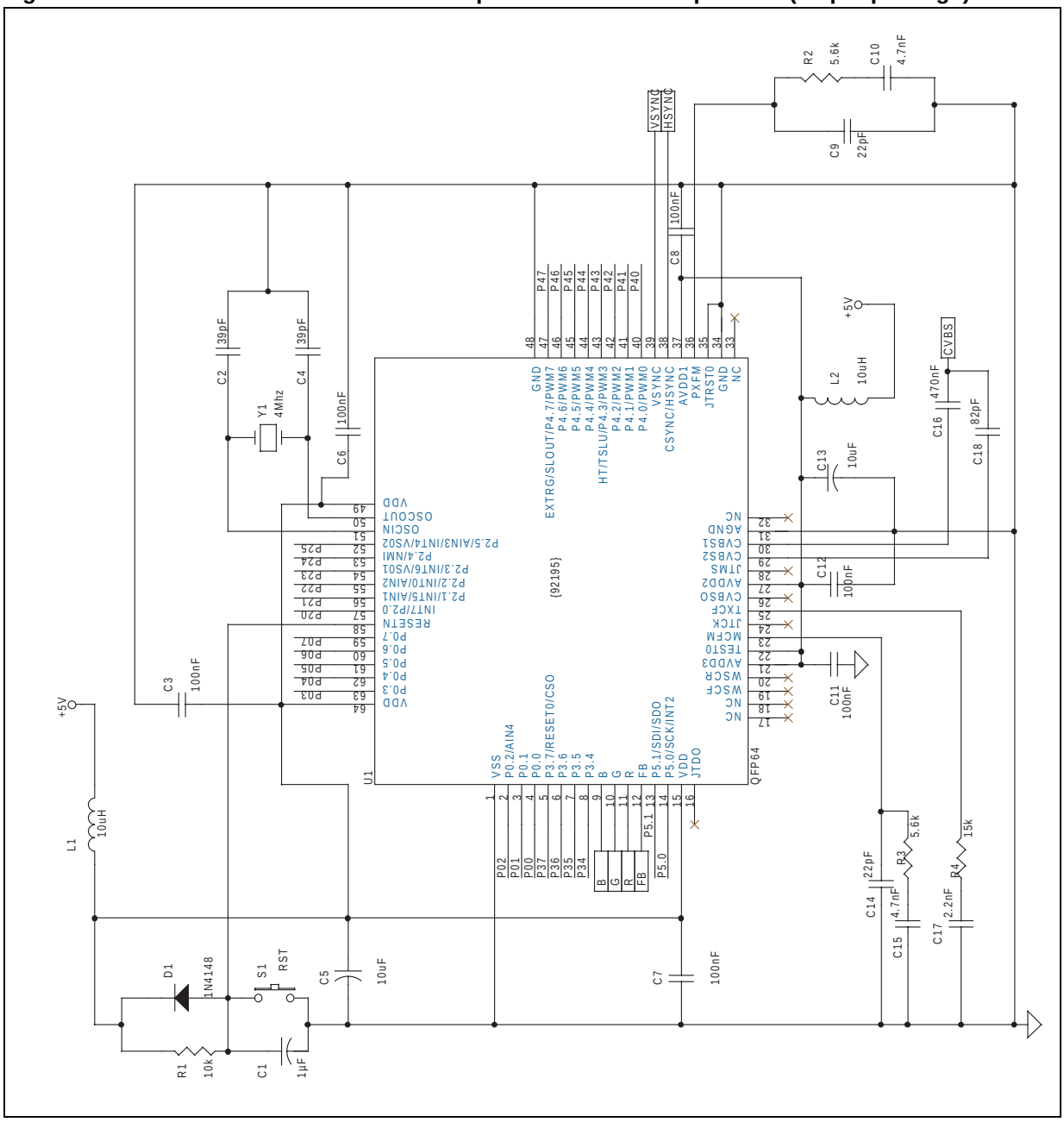
PIN DESCRIPTION (Cont'd)

Figure 120. ST92E195C/D-ST92T195C/D Required External components (56-pin package)



PIN DESCRIPTION (Cont'd)

Figure 121. ST92E195C/D-ST92T195C/D Required External Components (64-pin package)



PIN DESCRIPTION (Cont'd)

P0[7:0], P2[5:0], P3[7:4], P4[7:0], P5[1:0]
I/O Port Lines (Input/Output, TTL or CMOS compatible).

28 lines grouped into I/O ports, bit programmable as general purpose I/O or as Alternate functions (see I/O section).

Important: Note that open-drain outputs are for logic levels only and are not true open drain.

1.1.1 I/O Port Alternate Functions.

Each pin of the I/O ports of the ST92195C/D may assume software programmable Alternate Functions (see Table 1).

Table 37. ST92E195C/D and ST92T195C/D I/O Port Alternate Function Summary

Port Name	General Purpose I/O	Pin No.		Alternate Functions		
		CQFP64	CSDIP56			
P0.0	All ports useable for general purpose I/O (input, output or bidirectional)	4	10		I/O	
P0.1		3	9		I/O	
P0.2		2	8	AIN4	I	A/D Analog Data Input 4
P0.3		63	7		I/O	
P0.4		62	6		I/O	
P0.5		61	5		I/O	
P0.6		60	4		I/O	
P0.7		59	3		I/O	
P2.0		57	1	INT7	I	External Interrupt 7
P2.1		56	56	AIN1	I	A/D Analog Data Input 1
				INT5	I	External Interrupt 5
P2.2		55	55	INT0	I	External Interrupt 0
				AIN2	I	A/D Analog Data Input 2
P2.3		54	54	INT6	I	External Interrupt 6
				VSO1	O	Voltage Synthesis Output 1
P2.4		53	53	NMI	I	Non Maskable Interrupt Input
P2.5		52	52	AIN3	I	A/D Analog Data Input 3
				INT4	I	External Interrupt 4
				VSO2	O	Voltage Synthesis Output 2
P3.4		8	14		I/O	
P3.5		7	13		I/O	
P3.6		6	12		I/O	
P3.7		5	11	$\overline{\text{RESET0}}$	O	Internal Reset Output
				CSO	O	Composite Sync output
P4.0		40	42	PWM0	O	PWM Output 0
P4.1		41	43	PWM1	O	PWM Output 1
P4.2		42	44	PWM2	O	PWM Output 2
P4.3		43	45	PWM3	O	PWM Output 3
				TSLU	O	Translucency Digital Output
				HT	O	Half-tone Output

Port Name	General Purpose I/O	Pin No.		Alternate Functions		
		CQFP64	CSDIP56			
P4.4	All ports useable for general purpose I/O (input, output or bidirectional)	44	46	PWM4	O	PWM Output 4
				SCL2	I/O	I²C Channel 2 Serial Clock
P4.5		45	47	PWM5	O	PWM Output 5
				SDA2	I/O	I²C Channel 2 Serial Data
P4.6		46	48	PWM6	O	PWM Output 6
P4.7		47	49	EXTRG	I	A/D Converter External Trigger Input
				PWM7	O	PWM Output 7
				STOUT0	O	Standard Timer 0 Output
P5.0		14	20	INT2	I	External Interrupt 2
				SCK	O	SPI Serial Clock
				SCL1	I/O	I²C Channel 1 Serial Clock
P5.1		13	19	SDO	O	SPI Serial Data Out
				SDI	I	SPI Serial Data In
				SDA1	I/O	I²C Channel 1 Serial Data

1.1.2 I/O Port Styles

Pins	Weak Pull-Up	Port Style	Reset Values
P0[7:0]	no	Standard I/O	BID / OD / TTL
P2[5,4,3,2]	no	Standard I/O	BID / OD / TTL
P2[1,0]	no	Schmitt trigger	BID / OD / TTL
P3.7	yes	Standard I/O	AF / PP / TTL
P3[6,5,4]	no	Standard I/O	BID / OD / TTL
P4[7:0]	no	Standard I/O	BID / OD / TTL
P5[1:0]	no	Standard I/O	BID / OD / TTL

Legend:

AF= Alternate Function, BID = Bidirectional, OD = Open Drain
PP = Push-Pull, TTL = TTL Standard Input Levels

How to Read this Table

To configure the I/O ports, use the information in this table and the Port Bit Configuration Table in the I/O Ports Chapter on page 71.

Port Style= the hardware characteristics fixed for each port line.

Inputs:

– If port style = Standard I/O, either TTL or CMOS input level can be selected by software.

– If port style = Schmitt trigger, selecting CMOS or TTL input by software has no effect, the input will always be Schmitt Trigger.

Weak Pull-Up = This column indicates if a weak pull-up is present or not.

– If WPU = yes, then the WPU can be enabled/disable by software

– If WPU = no, then enabling the WPU by software has no effect

Alternate Functions (AF) = More than one AF cannot be assigned to an external pin at the same time:

An alternate function can be selected as follows.

AF Inputs:

– AF is selected implicitly by enabling the corresponding peripheral. Exception to this are ADC analog inputs which must be explicitly selected as AF by software.

PIN DESCRIPTION (Cont'd)

AF Outputs or Bidirectional Lines:

– In the case of Outputs or I/Os, AF is selected explicitly by software.

Example 1: ADC trigger digital input

AF: EXTRG, Port: P4.7, Port Style: Standard I/O.

Write the port configuration bits (for TTL level):

P4C2.7=1

P4C1.7=0

P4C0.7=1

Enable the ADC trigger by software as described in the ADC chapter.

Example 2: PWM 0 output

AF: PWM0, Port: P4.0

Write the port configuration bits (for output push-pull):

P4C2.0=0

P4C1.0=1

P4C0.0=1

Example 3: ADC analog input

AF: AIN1, Port : P2.1, Port style: does not apply to analog inputs

Write the port configuration bits:

P2C2.1=1

P2C1.1=1

P2C0.1=1

1.2 MEMORY MAP

Internal ROM

The ROM memory is mapped in two segments: segment 00h and segment 01h. It starts at address 0000h in MMU segment 00h.

Device	Size	Start Address	End Address
ST92T195D7 ST92E195D7	64K	00 0000h	00 FFFFh
ST92T195C8 ST92E195C8	84K	00 0000h	01 4FFFh

Internal RAM, 512 bytes

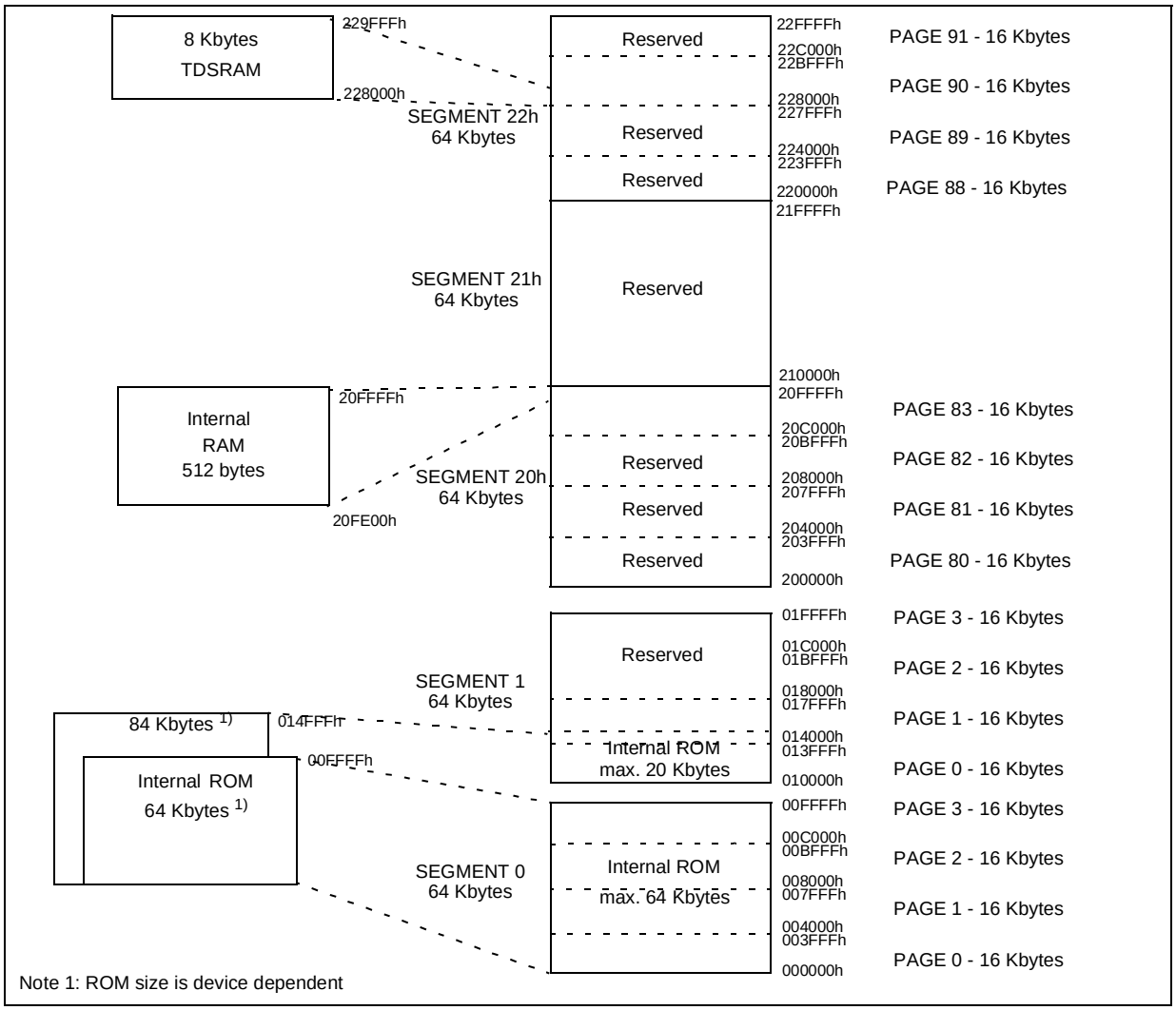
The internal RAM is mapped in MMU segment 20h; from address FE00h to FFFFh.

Internal TDSRAM

The Internal TDSRAM is mapped starting at address 8000h in MMU segment 22h. It is a fully static memory.

Device	Size	Start Address	End Address
ST92E195D7 ST92T195D7 ST92E195C8 ST92T195C8	8K	8000h	9FFFh

Figure 122. ST92E195C/D-ST92T195C/D Memory Map



Note 1: ROM size is device dependent

1.3 REGISTER MAP

The following pages contain a list of ST92E195C/D and ST92T195C/D registers, grouped by peripheral or function.

Be very careful to correctly program both:

– The set of registers dedicated to a particular function or peripheral.

– Registers common to other functions.

In particular, double-check that any registers with “undefined” reset values have been correctly initialised.

Warning: Note that in the **EIVR** and each **IVR** register, all bits are significant. Take care when defining base vector addresses that entries in the Interrupt Vector table do not overlap.

Group F Pages Register Map

Register	Page																	
	0	2	3	6	11	21	32	33	34	35	36	37	38	39	44	55	59	62
R255	Res.	Res.	Res.	VPS/ WSS	Res.	Res.	OSD		ACQ	TSU	TDS RAM			Res.	Res	RCCU (PLL)	VS	Res.
R254	SPI	Port 3												TCC				
R253																	Res.	
R252	WCR					MMU											PWM	
R251	WDT	Res.			STIM 1													
R250		Port 2												Res.		Res.		
R249																		
R248																		
R247	EXT INT	Res.			Res	Res.				Res.								
R246			Port 5			MMU												
R245															I²C			
R244																		
R243			Res.		STIM 0					SYNC								
R242		Port 0	Port 4															A/D
R241	Res.									Res.								
R240																		

Table 38. Detailed Register Map

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
N/A	I/O Port 0:5	R224	P0DR	Port 0 Data Register	FF	68
		R226	P2DR	Port 2 Data Register	FF	
		R227	P3DR	Port 3 Data Register	FF	
		R228	P4DR	Port 4 Data Register	FF	
		R229	P5DR	Port 5 Data Register	FF	
	Core	R230	CICR	Central Interrupt Control Register	87	56
		R231	FLAGR	Flag Register	00	29
		R232	RP0	Pointer 0 Register	xx	31
		R233	RP1	Pointer 1 Register	xx	31
		R234	PPR	Page Pointer Register	xx	33
		R235	MODER	Mode Register	E0	33
		R236	USPHR	User Stack Pointer High Register	xx	35
		R237	USPLR	User Stack Pointer Low Register	xx	35
		R238	SSPHR	System Stack Pointer High Reg.	xx	35
		R239	SSPLR	System Stack Pointer Low Reg.	xx	35
0	INT	R242	EITR	External Interrupt Trigger Register	00	56
		R243	EIPR	External Interrupt Pending Reg.	00	57
		R244	EIMR	External Interrupt Mask-bit Reg.	00	57
		R245	EIPLR	External Interrupt Priority Level Reg.	FF	57
		R246	EIVR	External Interrupt Vector Register	x6	58
		R247	NICR	Nested Interrupt Control	00	58
	WDT	R248	WDTHR	Watchdog Timer High Register	FF	80
		R249	WDTLR	Watchdog Timer Low Register	FF	80
		R250	WDTPR	Watchdog Timer Prescaler Reg.	FF	80
		R251	WDTCR	Watchdog Timer Control Register	12	80
		R252	WCR	Wait Control Register	7F	81
	SPI	R253	SPIDR	SPI Data Register	xx	182
		R254	SPICR	SPI Control Register	00	182
2	I/O Port 0	R240	P0C0	Port 0 Configuration Register 0	00	68
		R241	P0C1	Port 0 Configuration Register 1	00	
		R242	P0C2	Port 0 Configuration Register 2	00	
	I/O Port 2	R248	P2C0	Port 2 Configuration Register 0	00	
		R249	P2C1	Port 2 Configuration Register 1	00	
		R250	P2C2	Port 2 Configuration Register 2	00	
	I/O Port 3	R252	P3C0	Port 3 Configuration Register 0	00	
		R253	P3C1	Port 3 Configuration Register 1	00	
		R254	P3C2	Port 3 Configuration Register 2	00	

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
3	I/O Port 4	R240	P4C0	Port 4 Configuration Register 0	00	68
		R241	P4C1	Port 4 Configuration Register 1	00	
		R242	P4C2	Port 4 Configuration Register 2	00	
	I/O Port 5	R244	P5C0	Port 5 Configuration Register 0	00	
		R245	P5C1	Port 5 Configuration Register 1	00	
		R246	P5C2	Port 5 Configuration Register 2	00	
6	VPS/ WSS	R240	VPSSR	VPS Status Register	00	170
		R241	VPSD0R	VPS Data Register 0	00	170
		R242	VPSD1R	VPS Data Register 1	00	170
		R243	VPSD2R	VPS Data Register 2	00	170
		R244	VPSD3R	VPS Data Register 3	00	171
		R245	VPSD4R	VPS Data Register 4	00	171
		R246	WSSDS0R	WSS Data and Status Register 0	00	171
		R247	WSSDS1R	WSS Data and Status Register 1	00	171
		R248	WSSDS2R	WSS Data and Status Register 2	00	171
		R249	VPSWSSCR	VPS/WSS Control Register	00	172
		R250	WSSDS3R	WSS Data and Status Register 3	00	172
		R251	WSSDS4R	WSS Data and Status Register 4	00	173
		R252	WSSDS5R	WSS Data and Status Register 5	00	173
11	STIM0	R240	ST0HR	Counter High Byte Register	FF	85
		R241	ST0LR	Counter Low Byte Register	FF	85
		R242	ST0PR	Standard Timer Prescaler Register	FF	85
		R243	ST0CR	Standard Timer Control Register	14	85
	STIM1	R248	ST1HR	Counter High Byte Register	FF	85
		R249	ST1LR	Counter Low Byte Register	FF	85
		R250	ST1PR	Standard Timer Prescaler Register	FF	85
		R251	ST1CR	Standard Timer Control Register	14	85
21	MMU	R240	DPR0	Data Page Register 0	xx	40
		R241	DPR1	Data Page Register 1	xx	40
		R242	DPR2	Data Page Register 2	xx	40
		R243	DPR3	Data Page Register 3	xx	40
		R244	CSR	Code Segment Register	00	41
		R248	ISR	Interrupt Segment Register	xx	41
		R249	DMASR	DMA Segment Register	xx	41
	Ext.Mem.	R246	EMR2	External Memory Register 2	0F	59

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
32	OSD	R240	HBLANKR	Horizontal Blank Register	03	125
		R241	HPOSR	Horizontal Position Register	03	125
		R242	VPOSR	Vertical Position Register	00	125
		R243	FSCCR	Full Screen Color Control Register	00	126
		R244	HSCR	Header & Status Control Register	2A	127
		R245	NCSR	National Character Set Control Register	00	128
		R246	CHPOSR	Cursor Horizontal Position Register	00	129
		R247	CVPOSR	Cursor Vertical Position Register	00	129
		R248	SCLR	Scrolling Control Low Register	00	130
		R249	SCHR	Scrolling Control High Register	00	131
		R250	DCM0R	Display Control Mode 0 Register	00	133
		R251	DCM1R	Display Control Mode 1 Register	00	134
		R252	TDPR	TDSRAM Pointer Register	00	134
		R253	DE0R	Display Enable 0 Control Register	FF	135
		R254	DE1R	Display Enable 1 Control Register	FF	135
		R255	DE2R	Display Enable 2 Control Register	xF	135
33		R240	DCR	Default Color Register	70	136
		R241	CAPVR	Cursor Absolute Vertical Position Register	00	136
		R246	TDPPR	TDSRAM Page Pointer Register	x0	136
		R247	TDHSPR	TDSRAM Header/Status Pointer Register	x0	136
34	ACQ	R242	ACQAD1R	Acquisition Address Register 1	xx	163
		R243	ACQAD0R	Acquisition Address Register 0	xx	163
		R248	ACQPOR	Acquisition Page Open Register	00	158
		R249	ACQMLR	Acquisition Magazine Locked Register	00	158
		R250	ACQNHRR	Acquisition New Header Received Register	00	158
		R251	ACQPRR	Acquisition Packet Request Register	00	159
		R252	ACQTQMR	Acquisition Teletext Quality Measure Register	00	159
		R253	ACQHD2R	Acquisition Hamming Decoding Register 2	xx	160
		R254	ACQHD1R	Acquisition Hamming Decoding Register 1	xx	160
		R255	ACQHD0R	Acquisition Hamming Decoding Register 0	xx	160
35	SYNC	R242	SCCS0R	Sync Controller Control and Status Register 0	00	144
		R243	SCCS1R	Sync Controller Control and Status Register 1	00	145
	TSU	R248	TXSCR	Teletext Slicer Control Register	06	163
		R249	TXSLIR	Teletext Slicer Initialization Register	0B	164
		R255	PASR	Pre-Amplifier and ADC Selection Register	00	164

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
36	TDSRAM	R240 .. R255	BUF0 .. BUF15	40-byte buffer	xx .. xx	91
37		R240 .. R255	BUF16 .. BUF31		xx .. xx	
38		R240 .. R247	BUF32 .. BUF39		xx .. xx	
		R248	BUFC	TDSRAM Buffer Control Register	08	92
		R250	MTBSA1	Multi-byte Transfer Start Address Register 1	80	91
		R251	MTBSA0	Multi-byte Transfer Start Address Register 0	00	91
		R252	CONFIG	TDSRAM Interface Configuration Register	06	93
39	TCC	R251	PXCCR	PLL Clock Control Register	00	68
		R252	SLCCR	Slicer Clock Control Register	00	68
		R253	MCCR	Main Clock Control Register	00	67
		R254	SKCCR	Skew Clock Control Register	00	67
44	I²C	R240	I²COAR	Own Address Register	00	on page 189189
		R241	I²CFQR	Frequency Register	00	190
		R242	I²CCTR	Control Register	01	191
		R243	I²CDR	Data Register	00	192
		R244	I²CSTR2	Status Register 2	00	192
		R245	I²CSTR1	Status Register 1	00	193
55	RCCU	R251	PCONF	PLL Configuration Register	07	63
		R254	SDRATH	Clock Slow Down Unit Ratio Register	2x,4x or 00	63

Group F Page Dec.	Block	Reg. No.	Register Name	Description	Reset Value Hex.	Doc. Page
59	PWM	R240	CM0	Compare Register 0	00	207
		R241	CM1	Compare Register 1	00	207
		R242	CM2	Compare Register 2	00	207
		R243	CM3	Compare Register 3	00	207
		R244	CM4	Compare Register 4	00	207
		R245	CM5	Compare Register 5	00	207
		R246	CM6	Compare Register 6	00	207
		R247	CM7	Compare Register 7	00	207
		R248	ACR	Autoclear Register	FF	208
		R249	CCR	Counter Register	00	208
		R250	PCTL	Prescaler and Control Register	0C	208
		R251	OCPL	Output Complement Register	00	209
		R252	OER	Output Enable Register	00	209
	VS	R254	VSDR1	Data and Control Register 1	00	204
		R255	VSDR2	Data Register 2	00	204
62	ADC	R240	ADDTR	Channel i Data Register	xx	199
		R241	ADCLR	Control Logic Register	00	198
		R242	ADINT	AD Interrupt Register	01	199

Note: xx denotes a byte with an undefined value, however some of the bits may have defined values. Refer to register description for details.

2 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	V_{SS} - 0.3 to V_{SS} + 7.0	V
V_{SSA}	Analog Ground	V_{SS} - 0.3 to V_{SS} + 0.3	V
V_{DDA}	Analog Supply Voltage	V_{DD} -0.3 to V_{DD} +0.3	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} +0.3	V
V_{AI}	Analog Input Voltage (A/D Converter)	V_{SS} - 0.3 to V_{DD} +0.3 V_{SSA} - 0.3 to V_{DDA} +0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	- 55 to + 150	°C
I_{INJ}	Pin Injected Current Maximum Accumulated Pin Injected Current In Device	- 5 to + 5 - 50 to +50	mA

Note: Stress above those listed as “Absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Supply Voltage	4.5	5.5	V
V_{DDA}	Analog Supply Voltage (PLL)	4.5	5.5	V

Note: In order for the data slicer to operate correctly, the crystal frequency must be either 4 or 8 MHz. Please refer to Section 4, RESET AND CLOCK CONTROL UNIT (RCCU).

DC ELECTRICAL CHARACTERISTICS

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Value		Unit
			Min.	Max.	
V_{IHCK}	Clock In high level	External clock	0.7 V_{DD}		V
V_{ILCK}	Clock in low level	External clock		0.3 V_{DD}	V
V_{IH}	Input high level	TTL	2.0		V
V_{IL}	Input low level	TTL		0.8	V
V_{IH}	Input high level	CMOS	0.7 V_{DD}		V
V_{IL}	Input low level	CMOS		0.3 V_{DD}	V
V_{IHRS}	Reset in high level		0.7 V_{DD}		V
V_{ILRS}	Reset in low level			0.3 V_{DD}	V
V_{HYRS}	Reset in hysteresis		0.3		V
V_{IHY}	P2.(1:0) input hysteresis		0.9		V
V_{IHVH}	HSYNC/VSYNC input high level		0.7 V_{DD}		V
V_{ILVH}	HSYNC/VSYNC input low level			0.3 V_{DD}	V
V_{HYHV}	HSYNC/VSYNC input hysteresis		0.5		V
V_{OH}	Output high level	Push-pull I_{LOAD} =-0.8 mA	V_{DD}-0.8		V
V_{OL}	Output low level	Push-pull I_{LOAD} =+1.6 mA		0.4	V
I_{WPU}	Weak pull-up current	Bidirectional weak pull-up $V_{IN} = V_{SS}$	-50	-420	µA
I_{LKIO}	I/O pin input leakage current	$0<V_{IN}<V_{DD}$	-10	+10	µA
I_{LKRS}	Reset pin input	$0<V_{IN}<V_{DD}$	-10	+10	µA
I_{LKAD}	A/D pin input leakage current	alternate funct. op. drain	-10	+10	µA
I_{LKOS}	OSCIN pin input leakage current	$0<V_{IN}<V_{DD}$	-10	+10	µA

AC ELECTRICAL CHARACTERISTICS

PIN CAPACITANCE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Conditions	Value		Unit
			min	max	
C_{IO}	Pin Capacitance Digital Input/Output			10	pF

CURRENT CONSUMPTION

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	Conditions	Value			Unit
			min	typ.	max	
I_{DD1}	Run Mode Current	Notes 1,2; all ON		70	100	mA
I_{DDA1}	Run Mode Analog Current (pin V_{DDA})	Timing Controller ON		35	50	mA
I_{DD2}	HALT Mode Digital Current	Notes 1,4		10	100	µA
I_{DDA2}	HALT Mode Analog Current (pin V_{DDA})	Notes 1,4		10	100	µA

Notes:

1. Port 0 is configured in push-pull output mode (output is high). Ports 2, 3, 4 and 5 are configured in bi-directional weak pull-up mode resistor. The external CLOCK pin (OSCIN) is driven by a square wave external clock at 8 MHz. The internal clock prescaler is in divide-by-1 mode.

2. The CPU is fed by a 24 MHz frequency issued by the Main Clock Controller. VSYNC is tied to V_{SS}, HSYNC is driven by a 15625Hz clock. All peripherals working including Display.

3. The CPU is fed by a 24 MHz frequency issued by the Main Clock Controller. VSYNC is tied to V_{SS}, HSYNC is driven by a 15625Hz clock. The TDSRAM interface and the Slicers are working; the Display controller is not working.

4. VSYNC and HSYNC tied to V_{SS}. External CLOCK pin (OSCIN) is hold low. All peripherals are disabled.

EXTERNAL INTERRUPT TIMING TABLE

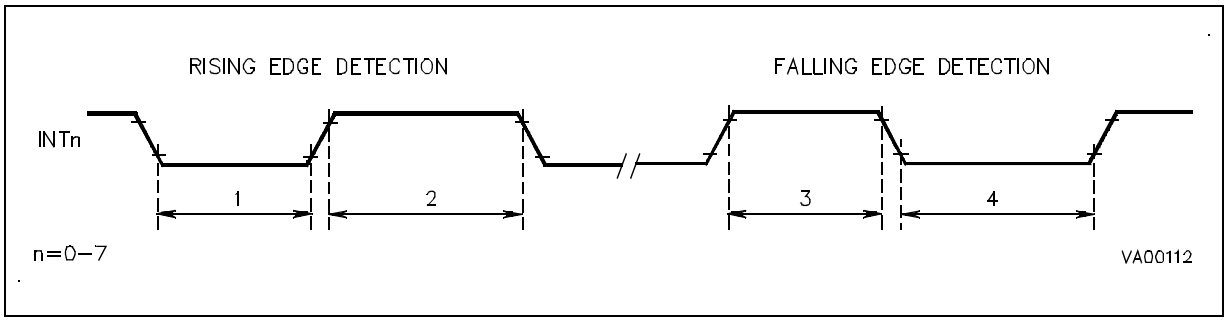
(V_{DD} = 5V ± 10%, T_A = 0°C +70° C, Cload = 50 pF, INTCLK = 12 MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Value (Note)			Unit
			OSCIN Divided by 2 Min.	OSCIN Not Divided by 2 Min.	Min.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95	ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95	ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95	ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95	ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.

TpC = OSCIN clock period

EXTERNAL INTERRUPT TIMING



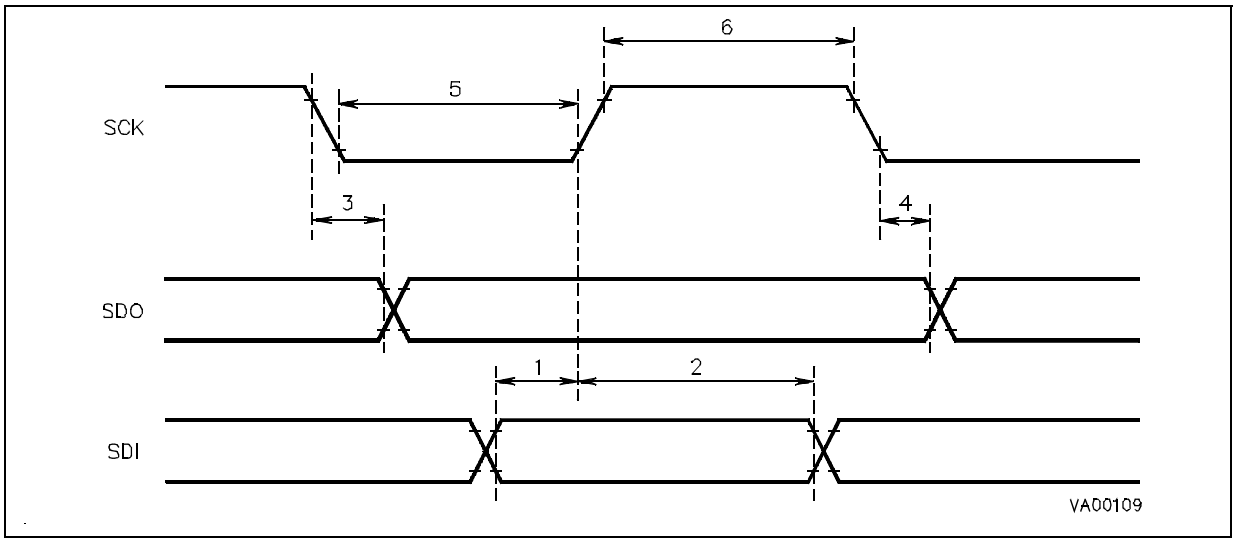
SPI TIMING TABLE

(V_{DD} = 5 V ± 10%, T_A = 0° C +70° C, Cload = 50 pF, INTCLK = 12 MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

SPI TIMING



SKEW CORRECTOR TIMING TABLE

(V_{DD}= 5V +/-10%, T_A= 0 to 70°C, unless otherwise specified)

Symbol	Parameter	Conditions	max Value	Unit
T_{jskw}	Jitter on RGB output	36 MHz Skew corrector clock frequency	5*	ns

(*) The OSD jitter is measured from leading edge to leading edge of a single character row on consecutive TV lines. The value is an envelope of 100 fields

OSD DAC CHARACTERISTICS

(V_{DD}= 5V +/-10%, T_A= 0 to 70°C, unless otherwise specified).

Symbol	Parameter	Conditions	Value			Unit
			min	typical	max	
	Output impedance: FB,R,G,B		300	500	700	Ohm
	Output voltage: FB,R,G,B	C_{LOAD} = 20 pF R_{LOAD} = 100 kΩ				
	code= 111			1.100	1.400	V
	code= 011			0.600	0.800	V
	code= 000			0.200	0.350	V
	FB= 1		V_{DD}-0.8			V
	FB= 0				0.400	V
	Global voltage accuracy				+/-5	%

AC ELECTRICAL CHARACTERISTICS (Cont'd)

A/D CONVERTER, EXTERNAL TRIGGER TIMING TABLE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Symbol	Parameter	OSCIN divide by 2;min/max	OSCIN divide by 1; min/max	Value		Unit
				min	max	
T_{low}	Pulse Width			1.5 INTCLK		ns
T_{ext}	Period/fast Mode			78+1 INTCLK		µs
T_{str}	Start Conversion Delay			0.5	1.5	INTCLK

A/D CONVERTER. ANALOG PARAMETERS TABLE

(V_{DD}= 5V +/-10%; T_A= 0 to 70°C; unless otherwise specified)

Parameter	Value			Unit	Note
	typ (*)	min	max	(**)	
Analog Input Range		V_{SS}	V_{DD}	V	
Conversion Time Fast/Slow		78/138		INTCLK	(1,2)
Sample Time Fast/Slow		51.5/87.5		INTCLK	(1)
Power-up Time		60		µs	
Resolution	8			bits	
Differential Non Linearity	1.5		2.5	LSBs	(4)
Integral Non Linearity	2		3	LSBs	(4)
Absolute Accuracy	2		3	LSBs	(4)
Input Resistance			1.5	Kohm	(3)
Hold Capacitance			1.92	pF	

Notes: (*)The values are expected at 25 Celsius degrees with V_{DD}= 5V
(**)'LSBs' , as used here, as a value of V_{DD}/256
(1)@ 24 MHz external clock
(2)including Sample time
(3)it must be considered as the on-chip series resistance before the sampling capacitor
(4)DNL ERROR= max {[V(i) -V(i-1)] / LSB-1}INL ERROR= max {[V(i) -V(0)] / LSB-i}
ABSOLUTE ACCURACY= overall max conversion error

I²C BUS TIMING

Symbol	Parameter	Standard I²C		Fast I²C		Unit
		Min	Max	Min	Max	
T_{BUF}	Bus free time between a STOP and START condition	4.7		1.3		ms
$T_{HD:STA}$	Hold time START condition. After this period, the first clock pulse is generated	4.0		0.6		µs
T_{LOW}	LOW period of the SCL clock	4.7		1.3		µs
T_{HIGH}	HIGH period of the SCL clock	4.0		0.6		µs
$T_{SU:STA}$	Set-up time for a repeated START condition	4.7		0.6		µs
$T_{HD:DAT}$	Data hold time	0 (1)		0 (1)	0.9(2)	ns
$T_{SU:DAT}$	Data set-up time	250		100		ns
T_R	Rise time of both SDA and SCL signals		1000	20+0.1Cb	300	ns
TF	Fall time of both SDA and SCL signals		300	20+0.1Cb	300	ns
$T_{SU:STO}$	Set-up time for STOP condition	4.0		0.6		ns
Cb	Capacitive load for each bus line		400		400	pF

1)The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL

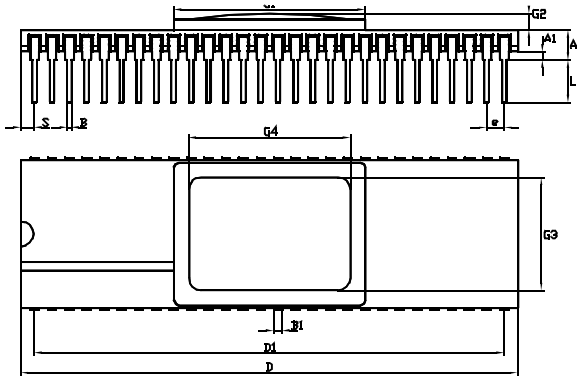
2)The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal

Cb = total capacitance of one bus line in pF

3 GENERAL INFORMATION

3.1 PACKAGE MECHANICAL DATA

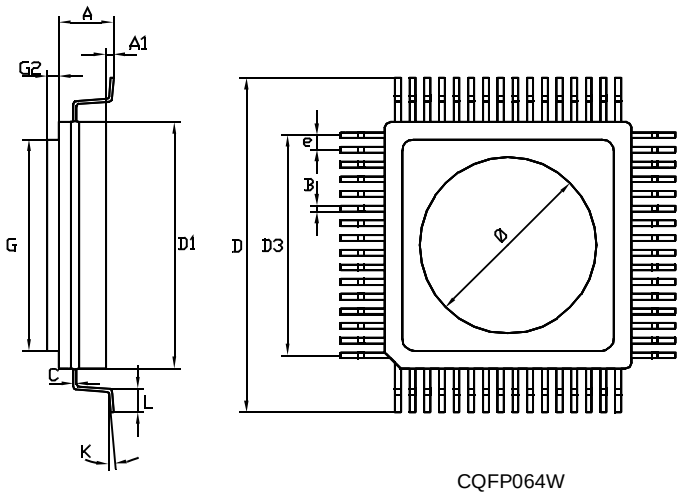
Figure 123. 56-Pin Shrink Ceramic Dual In-Line Package, 600-mil Width



CDIP56SW

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			4.17			0.164
A1	0.76			0.030		
B	0.38	0.46	0.56	0.015	0.018	0.022
B1	0.76	0.89	1.02	0.030	0.035	0.040
C	0.23	0.25	0.38	0.009	0.010	0.015
D	50.04	50.80	51.56	1.970	2.000	2.030
D1		48.01			1.890	
E1	14.48	14.99	15.49	0.570	0.590	0.610
e		1.78			0.070	
G	14.12	14.38	14.63	0.556	0.566	0.576
G1	18.69	18.95	19.20	0.736	0.746	0.756
G2		1.14			0.045	
G3	11.05	11.30	11.56	0.435	0.445	0.455
G4	15.11	15.37	15.62	0.595	0.605	0.615
L	2.92		5.08	0.115		0.200
S		1.40			0.055	
	Number of Pins					
N	56					

Figure 124. 64-Pin Ceramic Quad Flat Package

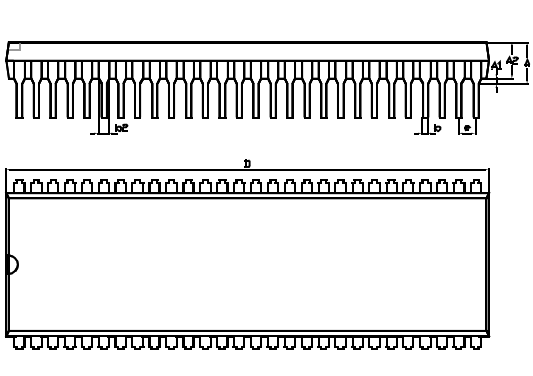


CQFP064W

Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.27			0.129
A1		0.50			0.020	
B	0.30	0.35	0.45	0.012	0.014	0.018
C	0.13	0.15	0.23	0.005	0.006	0.009
D	16.65	17.20	17.75	0.656	0.677	0.699
D1	13.57	13.97	14.37	0.534	0.550	0.566
D3		12.00			0.472	
e		0.80			0.031	
G		12.70			0.500	
G2		0.96			0.038	
L	0.35	0.80		0.014	0.031	
θ		8.31			0.327	
	Number of Pins					
N	64					

PACKAGE MECHANICAL DATA (Cont'd)

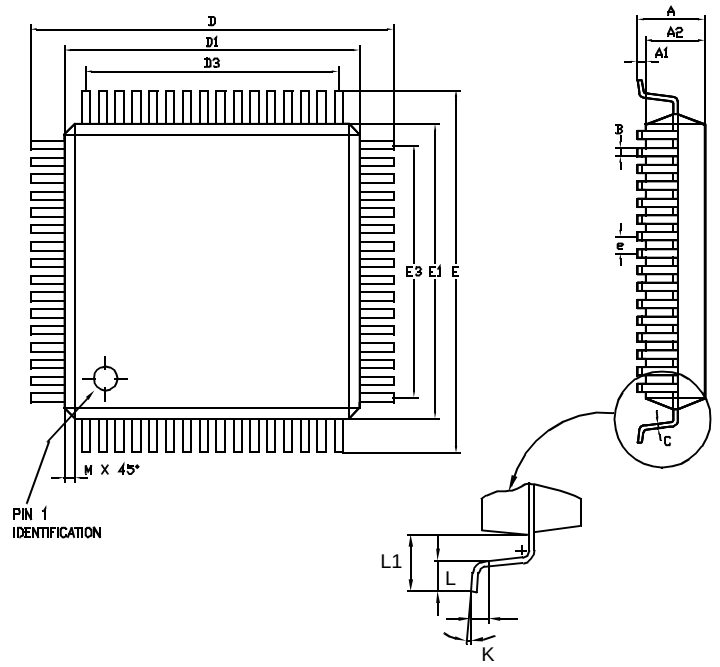
Figure 125. 56-Pin Shrink Plastic Dual In Line Package, 600-mil Width



PDIP56S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			6.35			0.250
A1	0.38			0.015		
A2	3.18		4.95	0.125		0.195
b		0.41			0.016	
b2		0.89			0.035	
C	0.20		0.38	0.008		0.015
D	50.29		53.21	1.980		2.095
E		15.01			0.591	
E1	12.32		14.73	0.485		0.580
e		1.78			0.070	
eA		15.24			0.600	
eB			17.78			0.700
L	2.92		5.08	0.115		0.200
	Number of Pins					
N	56					

Figure 126. 64-Pin Thin Quad Flat Package



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.30	0.37	0.45	0.012	0.015	0.018
C	0.09		0.20	0.004		0.008
D		16.00			0.630	
D1		14.00			0.551	
D3		12.00			0.472	
E		16.00			0.630	
E1		14.00			0.551	
E3		12.00			0.472	
e		0.80			0.031	
K	0°	3.5°	7°			
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	64		ND	16	NE	16

3.2 ORDERING INFORMATION

Each device is available for production in a user programmable version (OTP) as well as in factory coded version (ROM). OTP devices are shipped to customer with a default blank content FFh, while ROM factory coded parts contain the code sent by customer. The common EPROM versions for debugging and prototyping features the maximum memory size and peripherals of the family. Care must be taken to only use resources available on the target device.

3.2.1 Transfer Of OSD Customer Code

OSD Customer code is made up of the contents of 2 OSD files named xxx.OS0 and xxx.OS1.

Figure 127. OTP User Programmable Device Types

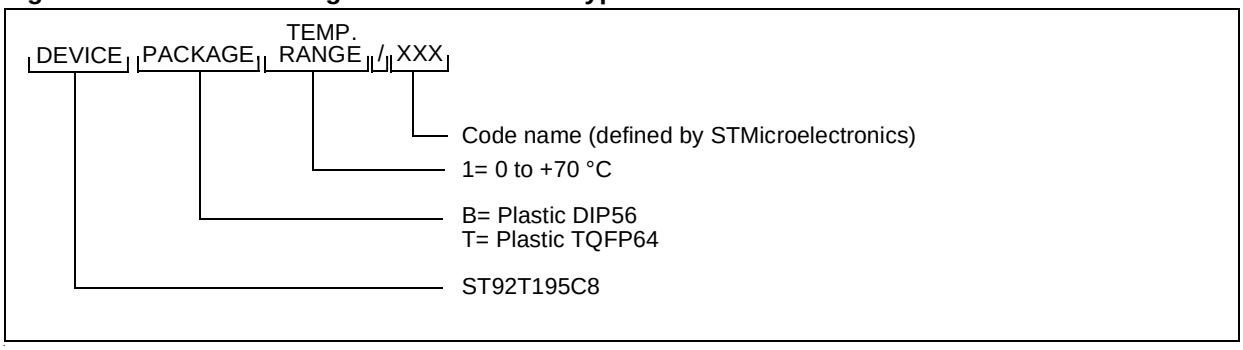
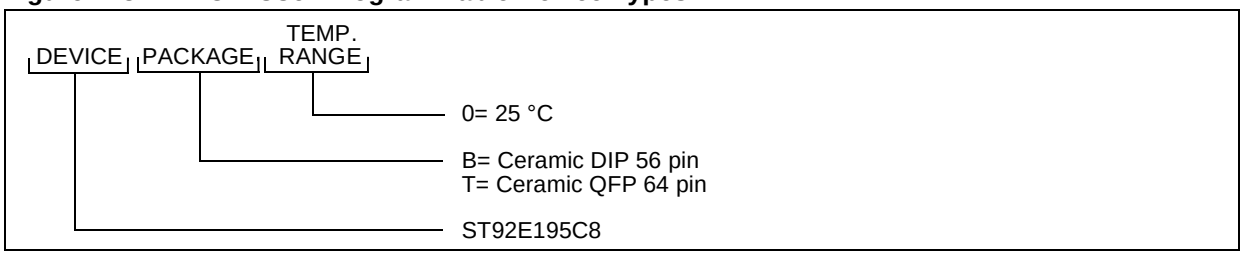


Figure 128. EPROM User Programmable Device Types



STMicroelectronics OPTION LIST

ST92T195C/D - ST92E195C/D

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Fax No: . . . . . .

STMicroelectronics reference:

Device: [ ] ST92E195C8 84K EPROM [ ] ST92E195D7 64K EPROM

[ ] ST92T185C8 84K OTP [ ] ST92T195D7 64K OTP

Package : [ ] SDIP56
[ ] QFP64

Temperature Range : 0° to 70° C

Software Development: [ ] STMicroelectronics
[ ] Customer
[ ] External laboratory

Special Marking:[ ] No SDIP56 [ ] Yes "_ _ _ _ _ _ _ _ _ _ _ _ _ _"
QFP64 [ ] Yes "_ _ _ _ _ _ _ _ _ _ "

For marking, one line is possible with maximum 14 characters (SDIP56) or 10 characters (TQFP64). Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Please consult your local ST Microelectronics sales office for other marking details if required.

Notes :

OSD Code : [ ] OSD Files Filename [..... .OS0]Filename [..... .OS1]
Quantity forecast : [.....] k units per year
For a period of : [.....] years
Preferred Production start dates : [../../..] (YY/MM/DD)

Date
Customer Signature :

4 REVISION HISTORY

Rev.	Main Changes	Date
1.0	First release on DMS.	07/12/00
1.1	From p182, changed I2C cell: - removed references to 4 buses - changed last sentence (introduction) p9 and p221, added SDA1 and SCL1 in figure 5 and 116. p153: changed 7.7.5 section p208, changed value (4 or 8 MHz instead of 3.3 or 8.7 MHz) in the second note. p219, added note in paragraph 1.0.5. added one option list added PSDIP56 and TQFP64 package for the EPROM/OTP section	08/18/00
1.2	Changed option list.	09/29/00
1.3	Addition of Section 4 on page 248. Modified Table 11. on page 61. Updated Section 7.4.5.1 on page 114 and Section 4.2 on page 60. Updated Figure 4, Figure 5, Figure 120 and Figure 121.	11 Oct 2001

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

©2003 STMicroelectronics - All Rights Reserved.

Purchase of I^2C Components by STMicroelectronics conveys a license under the Philips I^2C Patent. Rights to use these components in an I^2C system is granted provided that the system conforms to the I^2C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies
Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com