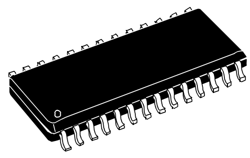
ST7FLCD1

8-bit MCU for LCD Monitors with 60 KBytes Flash, 2 KBytes RAM, 2 DDC Ports and Infrared Controller

Key Features

- **60 KBytes Flash Program Memory**
- **In-Circuit Debugging and Programming**
- **In-Application Programming**
- **Data RAM: up to 2 KBytes (256 bytes stack, 2 x 256 bytes for DDCs)**
- **8 MHz, up to 9 MHz Internal Clock Frequency**
- **True Bit Manipulation**
- **Run and Wait CPU Modes**
- **Programmable Watchdog for System Reliability**
- **Protection against Illegal Opcode Execution**
- **2 DDC Bus Interfaces with:**
  - DDC 2B protocol implemented in hardware
  - Programmable DDC CI modes
  - Enhanced DDC (EDDC) address decoding
  - HDCP Encryption keys
- **Fast I^2C Single Master Interface**
- **8-bit Timer with Programmable Pre-scaler, Auto-reload and independent Buzzer Output**
- **8-bit Timer with External Trigger**
- **4-channel, 8-bit Analog to Digital Converter**
- **4 + 2 8-bit PWM Digital to Analog Outputs with Frequency Adjustment**
- **Infrared Controller (IFR)**
- **Up to 22 I/O Lines in 28-pin Package**
- **2 Lines Programmable as Interrupt Inputs**
- **Master Reset and Low Voltage Detector (LVD) Reset**
- **Complete Development Support on PC-Windows**
- **Full Software Package (Assembler, Linker, C-compiler and Source Level Debugger)**



SO28
ORDER CODE: ST7FLCD1

General Description

The ST7FLCD1 is a microcontroller (MCU) from the ST7 family with dedicated peripherals for LCD monitor applications. The ST7FLCD1 is an industry standard 8-bit core that offers an enhanced instruction set. The 5V supplied processor runs with an external clock at 24 MHz (27 MHz maximum). Under software control, the MCU mode changes to Wait mode thus reducing power consumption. The enhanced instruction set and addressing modes offer real programming potential.

In addition to standard 8-bit data management, the MCU features also include true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

The device gathers the on-chip oscillator, CPU, 60-Kbyte Flash, 2-KByte RAM, I/Os, two 8-bit timers, infrared preprocessor, 4-channel Analog-to-Digital Converter, 2 DDCs, I^2C single master, watchdog, reset and six 8-bit PWM outputs for analog DC control of external functions.

Table of Contents

Chapter 1 General Information 6
1.1 Block Diagram 6
1.2 Abbreviations 6
1.3 Reference Documents 7
1.4 Pin Description 8
1.5 External Connections 10
1.6 Memory Map 11

Chapter 2 Central Processing Unit (CPU) 15
2.1 Main Features 15
2.1.1 CPU Registers 15

Chapter 3 Reset 22
3.1 Low Voltage Detector and Watchdog Reset 22
3.2 Watchdog or Illegal Opcode Access Reset 23
3.3 External Reset 23
3.4 Reset Procedure 23

Chapter 4 Interrupts 24
4.1 Software 24
4.2 External Interrupts (ITA, ITB) 24
4.3 Peripheral Interrupts 24
4.4 Processing 24
4.5 Register Description 26

Chapter 5 Flash Program Memory 28
5.1 Introduction 28
5.2 Main Features 28
5.3 Structure 28
5.4 Program Memory Read-out Protection 28
5.5 In-Circuit Programming (ICP) 29
5.6 In-Application Programming (IAP) 30
5.7 Register Description 30
5.8 Flash Option Bytes 31

Chapter 6 Clocks & Low Power Modes 32
6.1 Clock System 32
6.1.1 General Description 32
6.1.2 Crystal Oscillator Mode 32
6.1.3 External Clock Mode 32
6.1.4 Clock Signals 33
6.2 Power Saving Modes 33
6.2.1 HALT Mode 33
6.2.2 WAIT Mode 33
6.2.3 Exit from HALT and WAIT Modes 33
6.2.4 Selected Peripherals Mode 34

Chapter 7 I/O Ports 35
7.1 Introduction 35
7.2 Common Functional Description 36
7.3 Port A 37
7.4 Port B 39
7.5 Port C 40
7.6 Port D 41
7.7 Register Description 42

Chapter 8 PWM Generator 43
8.1 Introduction 43
8.2 Main Features 43
8.3 Functional Description 43
8.4 Register Description 46

Chapter 9 8-bit Analog-to-Digital Converter (ADC) 49
9.1 Introduction 49
9.2 Main Features 49
9.3 Functional Description 49
9.4 Register Description 50

Chapter 10 I²C Single-Master Bus Interface 52
10.1 Introduction 52
10.2 Main Features 52
10.3 General Description 52
10.4 Functional Description (Master Mode) 54
10.5 Transfer Sequencing 54

10.5.1 Master Receiver 54
10.5.2 Master Transmitter 54
10.6 Register Description 56

Chapter 11 Display Data Channel Interfaces (DDC) 60
11.1 Introduction 60
11.2 DDC Interface Features 60
11.2.1 Hardware DDC2B Interface Features 60
11.2.2 DDC/CI Factory Interface Features 60
11.3 Signal Description 62
11.3.1 Serial Data (SDA) 62
11.3.2 Serial Clock (SCL) 62
11.4 DDC Standard 62
11.4.1 DDC2B Interface 62
11.4.2 Mode Description 63
11.5 DDC/CI Factory Alignment Interface 66
11.5.1 I²C Modes 66
11.6 Transfer Sequencing 68
11.7 Register Description 69

Chapter 12 Watchdog Timer (WDG) 75
12.1 Introduction 75
12.2 Main Features 75
12.3 Main Watchdog Counter 75
12.4 Lock-up Counter 76
12.5 Interrupts 76
12.6 Register Description 76

Chapter 13 8-bit Timer (TIMA) 77
13.1 Introduction 77
13.2 Main Features 77
13.3 Functional Description 77
13.4 Register Description 78

Chapter 14 8-bit Timer with External Trigger (TIMB) 80
14.1 Introduction 80
14.2 Main Features 80
14.3 Functional Description 80

Chapter 15 Infrared Preprocessor (IFR) 83

15.1 Main Features 83

15.2 Functional Description 83

15.3 Register Description 84

Chapter 16 Registers 85

16.1 Register Description 85

Chapter 17 Electrical Characteristics 87

17.1 Absolute Maximum Ratings 87

17.2 Power Considerations 87

17.3 Thermal Characteristics 88

17.4 AC/DC Electrical Characteristics 88

17.5 Power On/Off Electrical Specifications 90

17.6 8-bit Analog-to-Digital Converter 90

17.7 I2C/DDC Bus Electrical Specifications 91

17.8 I2C/DDC Bus Timings 91

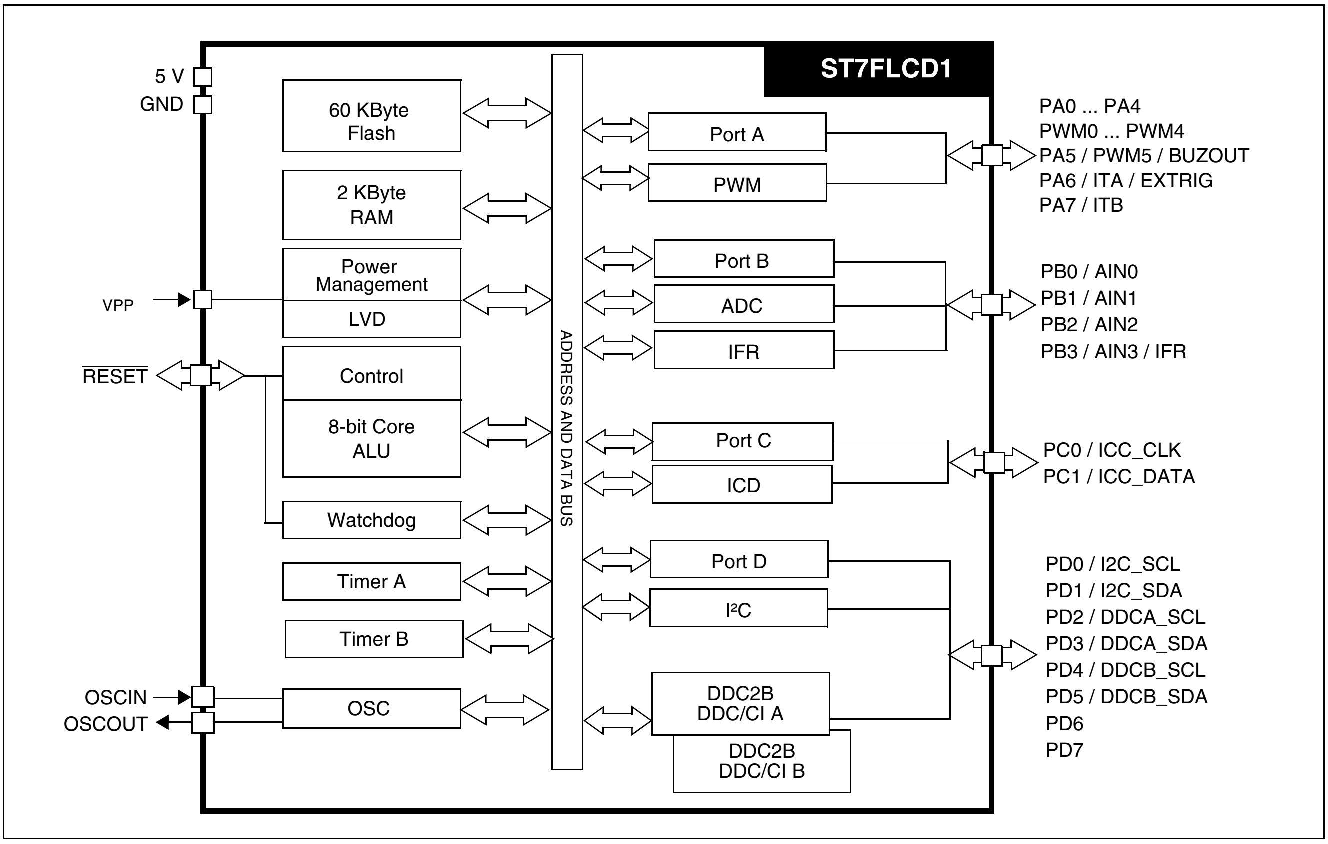
Chapter 18 Package Mechanical Data 93

Chapter 19 Revision History 94

1 General Information

1.1 Block Diagram

Figure 1: ST7FLCD1 Functional Diagram



1.2 Abbreviations

Abbreviation	Description
ADC	Analog-to-Digital Converter
ALU	Arithmetical and Logical Unit
CPU	Central Processing Unit
DDC	Display Data Channel
DMA	Direct Memory Access
I²C or IIC	Inter-Integrated Circuit bus
IAP	In-Application Programming
ICC	In-Circuit Communication
ICP	In-Circuit Programming
ICT	In-Circuit Testing
IFR	Infrared Controller

Abbreviation	Description
IT	Interrupt
LCD	Liquid Crystal Display
LVD	Low Voltage Detector
MCU	Microcontroller Unit
OSC	Oscillator
PWM	Pulse Width Modulator
TIM	Timer
WDG	Watchdog

1.3 Reference Documents

Book: ST7 MCU Family Manual

CD: MCU on CD

Many libraries, software and applications notes are available.

Ask your STMicroelectronics sales office, your local support or search the company web site at www.st.com

1.4 Pin Description

Figure 2: 28-pin Small Outline Package (SO28) Pinout

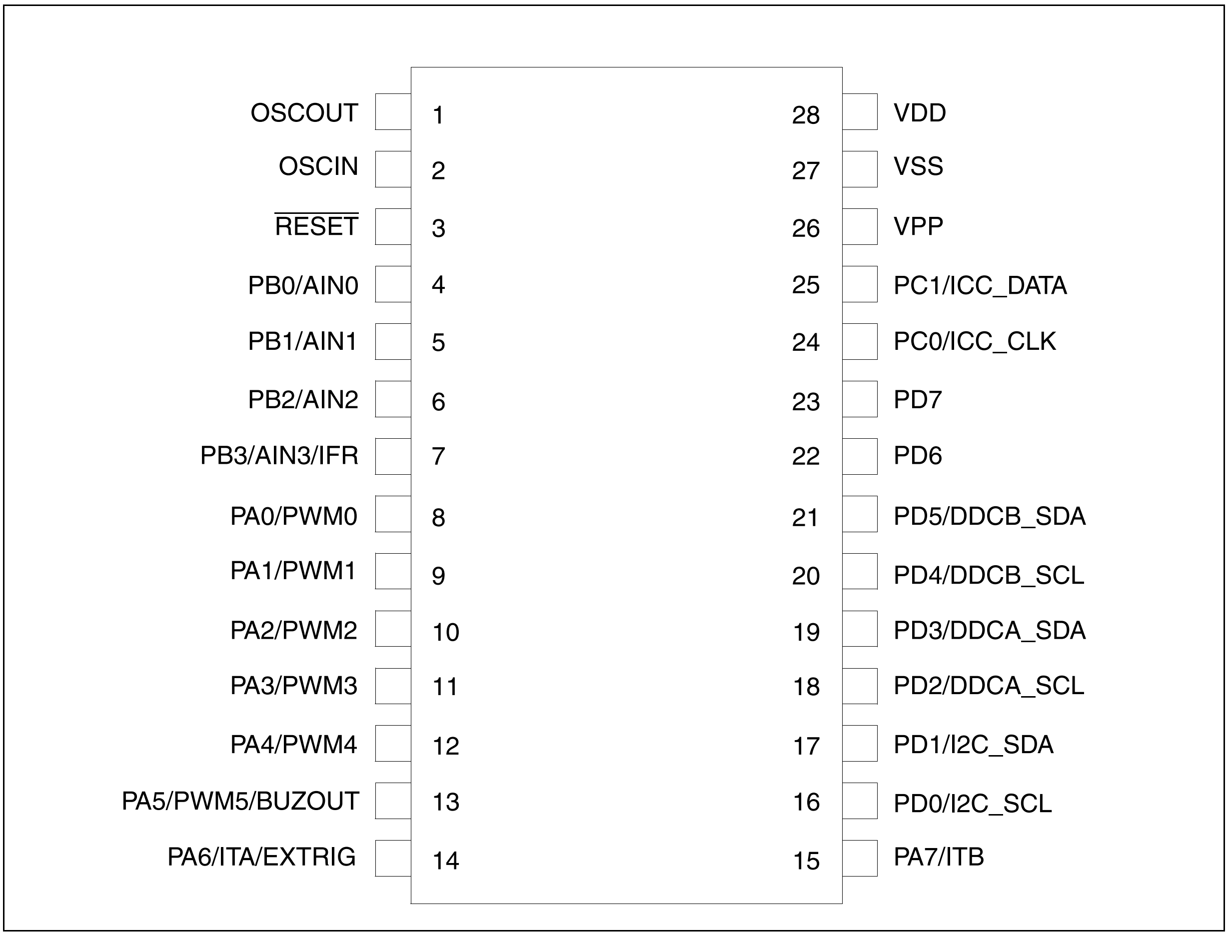


Table 1: 28-pin Small Outline Package (SO28) Pin Description (Sheet 1 of 2)

Pin	Pin Name	Type	Description	Remark
1	OSCOUT	O	Oscillator Input	Normal use at 24 MHz
2	OSCIN	I	Oscillator Output	
3	RESET	I/O	Reset	
4	PB0/AIN0	I/O	Port B0 or ADC Analog Input 0	
5	PB1/AIN1	I/O	Port B1 or ADC Analog Input 1	
6	PB2/AIN2	I/O	Port B2 or ADC Analog Input 2	
7	PB3/AIN3/IFR	I/O	Port B3 or ADC Analog Input 3 or IFR Input	
8	PA0/PWM0	I/O	Port A0 or PWM Output 0	
9	PA1/PWM1	I/O	Port A1 or PWM Output 1	
10	PA2/PWM2	I/O	Port A2 or PWM Output 2	
11	PA3/PWM3	I/O	Port A3 or PWM Output 3	
12	PA4/PWM4	I/O	Port A4 or PWM Output 4	

Table 1: 28-pin Small Outline Package (SO28) Pin Description (Sheet 2 of 2)

Pin	Pin Name	Type	Description	Remark
13	PA5/PWM5/BUZOUT	I/O	Port A5 or PWM Output 5 or Buzzer Output	
14	PA6/ITA/EXTRIG	I/O	Port A6 or Interrupt Input A or External Trigger Timer B	
15	PA7/ITB	I/O	Port A7 or Interrupt Input B	
16	PD0/I2C_SCL	I/O	Port D0 or I^2C Serial Bus Clock	
17	PD1/I2C_SDA	I/O	Port D1 or I^2C Serial Bus Data	
18	PD2/DDCA_SCL	I/O	Port D2 or DDC A Serial Bus Clock	
19	PD3/DDCA_SDA	I/O	Port D3 or DDC A Serial Bus Data	
20	PD4/DDCB_SCL	I/O	Port D4 or DDC B Serial Bus Clock	
21	PD5/DDCB_SDA	I/O	Port D5 or DDC B Serial Bus Data	
22	PD6	I/O	Port D6	
23	PD7	I/O	Port D7	
24	PC0/ICC_CLK	I/O	Port C0 or ICC Clock	
25	PC1/ICC_DATA	I/O	Port C1 or ICC Data	
26	VPP	PS	Flash Programming Supply Voltage	Normal op. mode: 0 V, see Note [1]
27	VSS	PS	Ground	0V
28	VDD	PS	Power Supply	5V

1. This pin must be connected to a 10K pulldown resistor (refer to Section 1.5).

1.5 External Connections

Figure 3 shows the recommended external connections for the device.

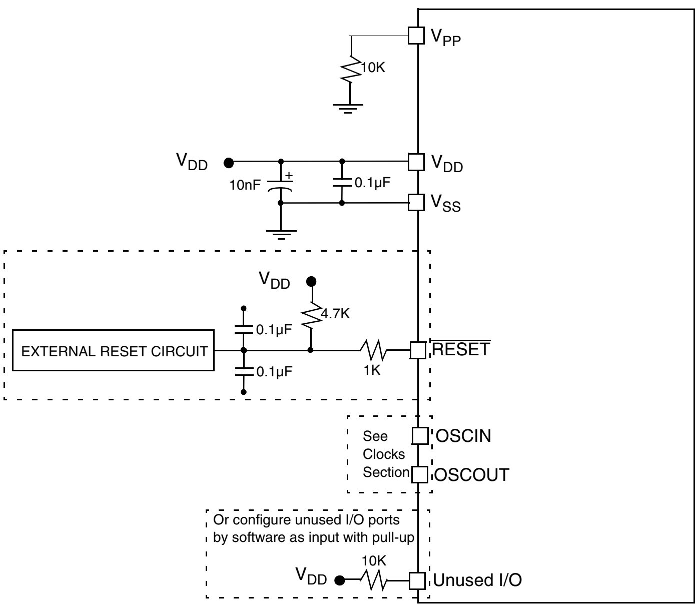
The V_{PP} pin is only used for programming or erasing the Flash memory array, and must be tied to a 10 K pulldown resistor for normal operation.

The 10 nF and 0.1 µF decoupling capacitors on the power supply lines are a suggested EMC performance/cost tradeoff.

The external RC reset network (including the mandatory 1K serial resistor) is intended to protect the device against parasitic resets, especially in noisy environments.

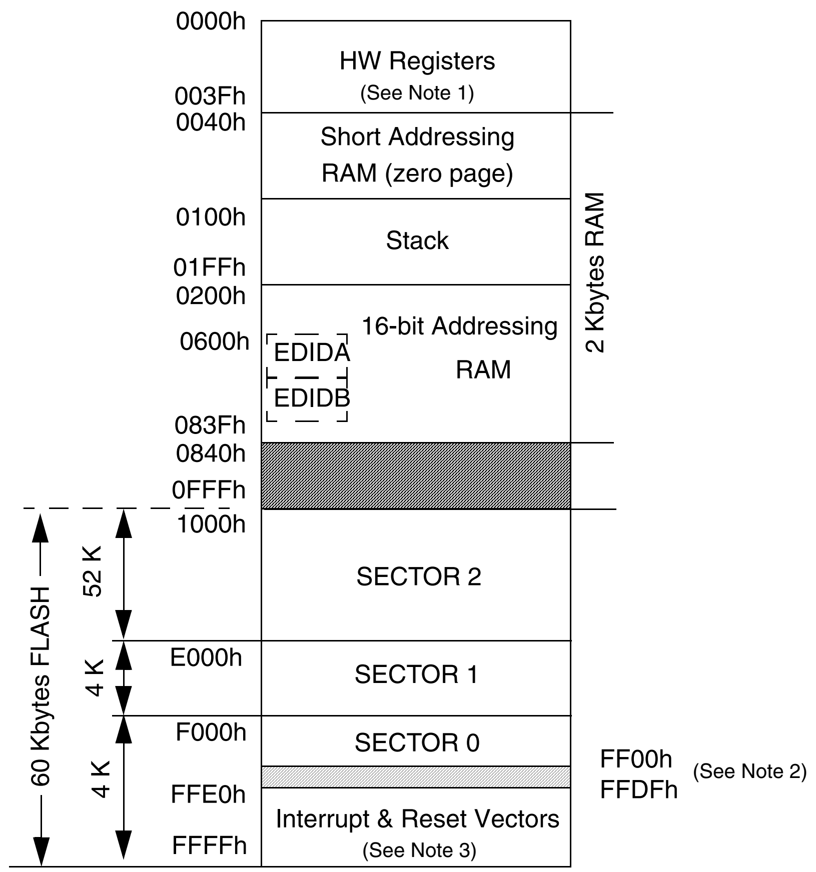
Unused I/Os should be tied high to avoid any unnecessary power consumption on floating lines. An alternative solution is to program the unused ports as inputs with pull-up.

Figure 3: Recommended External Connections



1.6 Memory Map

Figure 4: Program Memory Map



Note:1. Refer to Table 2: Hardware Register Memory Map.

2. Area FF00h to FFDFh is reserved in the event of ICD use. (For more information, refer to Application Note 1581.)

3. Refer to Table 3: Interrupt Vector Map.

Table 2: Hardware Register Memory Map (Sheet 1 of 3)

Address	Block	Register	Register Name	Reset Status	Remarks
0000h	NAME	NAMER	Circuit Name Register	00h	Read
0001h	MISC	MISCR	Miscellaneous Register	00h	R/W
0002h	Port A	PADR	Port A Data Register	00h	R/W
0003h		PADDR	Port A Data Direction Register	00h	R/W
0004h	Port B	PBDR	Port B Data Register	00h	R/W
0005h		PBDDR	Port B Data Direction Register	00h	R/W
0006h	Port C	PCDR	Port C Data Register	00h	R/W
0007h		PCDDR	Port C Data Direction Register	00h	R/W

Table 2: Hardware Register Memory Map (Sheet 2 of 3)

Address	Block	Register	Register Name	Reset Status	Remarks
0008h	Port D	PDDR	Port D Data Register	00h	R/W
0009h		PDDDR	Port D Data Direction Register	00h	R/W
000Ah	ADC	ADCDR	ADC Data Register	00h	R
000Bh		ADCCSR	ADC Control Status Register	00h	R/W
000Ch	INTERRUPT	ITRFRE	External Interrupt Register	00h	R/W
000Dh	TIMA	TIMCSRA	Timer Control Status Register	00h	R/W
000Eh		TIMCPRA	Timer Counter Preload Register	00h	R/W
000Fh	PWM	PWMDCR0	8-bit PWM0 Duty Cycle Register	00h	R/W
0010h		PWMDCR1	8-bit PWM1 Duty Cycle Register	00h	R/W
0011h		PWMDCR2	8-bit PWM2 Duty Cycle Register	00h	R/W
0012h		PWMDCR3	8-bit PWM3 Duty Cycle Register	00h	R/W
0013h		PWMCRA	PWM[0...3] Control Register	00h	R/W
0014h		PWMARRA	PWM[0...3] Auto Reload Register	FFh	R/W
0015h		PWMDCR4	8-bit PWM4 Duty Cycle Register	00h	R/W
0016h		PWMDCR5	8-bit PWM5 Duty Cycle Register	00h	R/W
0017h		PWMCRB	PWM[4...5] Control Register	00h	R/W
0018h		PWMARRB	PWM[4...5] Auto Reload Register	FFh	R/W
0019h	FLASH	FCSR	Flash Control/Status Register	00h	R/W
001Ah	Reserved				
001Bh	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
001Ch	I²C	I2CCR	I²C Control Register	00h	R/W
001Dh		I2CSR	I²C Status Register	00h	R
001Eh		I2CCCR	I²C Clock Control Register	00h	R/W
001Fh		I2CDR	I²C Data Register	00h	R/W
0020h	DDC A	DDCCRA	DDC A Control Register	00h	R/W
0021h		DDCSR1A	DDC A Status 1 Register	00h	R
0022h		DDCSR2A	DDC A Status 2 Register	00h	R
0023h		DDCOAR1A	DDC (7-bit) A Slave address 1 Register	00h	R/W
0024h		DDCOAR2A	DDC (7-bit) A Slave address 2 Register	00h	R/W
0025h		DDCDRA	DDC A Data Register	00h	R/W
0026h		RESERVED			
0027h		DDCDCRA	DDC2B A Control Register	00h	R/W

Table 2: Hardware Register Memory Map (Sheet 3 of 3)

Address	Block	Register	Register Name	Reset Status	Remarks
0028h	DDC B	DDCCRB	DDC B Control Register	00h	R/W
0029h		DDCSR1B	DDC B Status 1 Register	00h	R
002Ah		DDCSR2B	DDC B Status 2 Register	00h	R
002Bh		DDCOAR1B	DDC (7-bit) B Slave address 1 Register	00h	R/W
002Ch		DDCOAR2B	DDC (7-bit) B Slave address 2 Register	00h	R/W
002Dh		DDCDRB	DDC B Data Register	00h	R/W
002Eh		RESERVED			
002Fh		DDCDCRB	DDC2B B Control Register	00h	R/W
0030h	DM	DMCR	Debug Control Register	00h	R/W
0031h		DMSR	Debug Status Register	10h	R
0032h		DMBK1H	Debug Breakpoint 1 MSB Register	FFh	R/W
0033h		DMBK1L	Debug Breakpoint 1 LSB Register	FFh	R/W
0034h		DMBK2H	Debug Breakpoint 2 MSB Register	FFh	R/W
0035h		DMBK2L	Debug Breakpoint 2 LSB Register	FFh	R/W
0036h	IFR	IFRDR	Counter Data Register	00h	R/W
0037h		IFRCR	Control Register	00h	R/W
0038h	TIMB	TIMCSRB	Timer Control Status Register	00h	R/W
0039h		TIMCPRB	Timer Counter Preload Register	01h	R/W
003Ah	RESERVED				

Table 3: Interrupt Vector Map

Vector Address	Description	Remarks
FFE0 to FFE1h	Not Used	
FFE2 to FFE3h	Timer A Overflow Interrupt Vector	Internal Interrupt
FFE4 to FFE5h	Timer B Overflow Interrupt Vector	Internal Interrupt
FFE6 to FFE7h	Not Used	
FFE8 to FFE9h	I²C Interrupt Vector	Internal Interrupt
FFEA to FFEBh	ITB Interrupt Vector	External Interrupt
FFEC to FFEDh	ITA Interrupt Vector	External Interrupt
FFEE to FFEFh	IFR Interrupt Vector	Internal Interrupt
FFF0 to FFF1h	Not Used	
FFF2 to FFF3h	DDC2B B Interrupt Vector	Internal Interrupt
FFF4 to FFF5h	DDC/CI B Interrupt Vector	Internal Interrupt
FFF6 to FFF7h	DDC2B A Interrupt Vector	Internal Interrupt
FFF8 to FFF9h	DDC/CI A Interrupt Vector	Internal Interrupt
FFFA to FFFBh	Not Used	

Table 3: Interrupt Vector Map

Vector Address	Description	Remarks
FFFC to FFFDh	Trap (Software) Interrupt Vector	CPU Interrupt
FFFE to FFFFh	Reset Vector	

2 Central Processing Unit (CPU)

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

2.1 Main Features

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- 8 MHz CPU internal frequency (9 MHz maximum)
- Wait and Halt Low Power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

2.1.1 CPU Registers

The 6 CPU registers shown in Figure 5 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register that holds operands and results of arithmetic and logic calculations. It also manipulates data.

Index Registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a previous instruction (PRE) to indicate that next instruction refers to the Y register.)

The Y register is not affected by interrupt automatic procedures (not pushed to and popped from the stack).

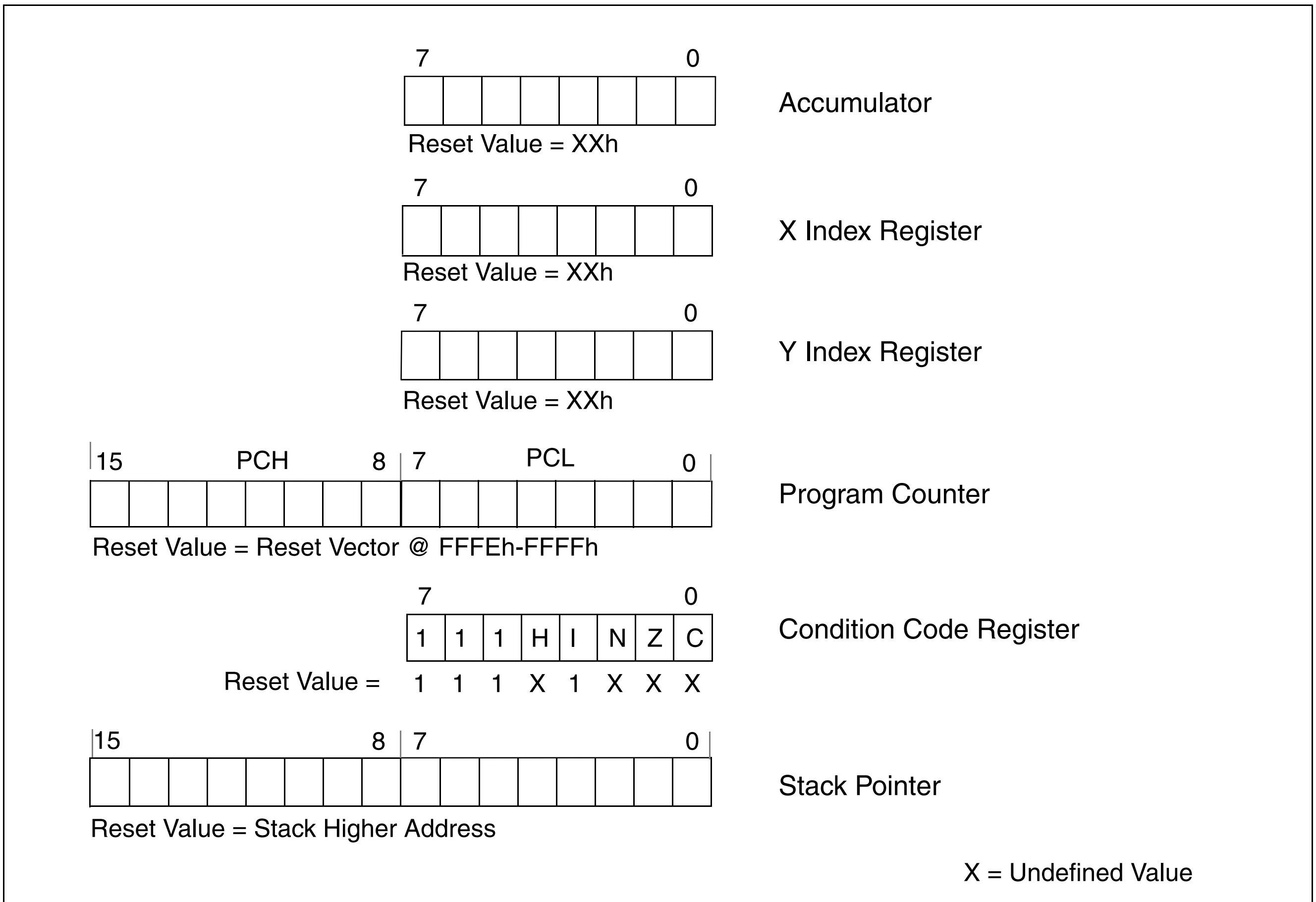
Program Counter (PC)

The program counter is a 16-bit register containing the address of next instruction the CPU executes. The program counter consists of two 8-bit registers:

PCL (Program Counter Low which is the LSB)

PCH (Program Counter High which is the MSB).

Figure 5: CPU Registers



CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1XXX

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags resulting from the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = H *Half carry.*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Note: Instruction Groups are defined in Table 5.

Bit 3 = I ***Interrupt mask.***

This bit is set by hardware by an interrupt or by software that disables all interrupts except the TRAP software interrupt. This bit is cleared by software.

0: Interrupts are enabled.

1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Interrupts requested when the I bit is set are latched and processed when the I bit is cleared. By default an interrupt routine is not interruptible as the I bit is set by hardware when you enter it and reset by the IRET instruction at the end of interrupt routine. In case the I bit is cleared by software during the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = N ***Negative.***

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.

0: The last operation result is positive or null.

1: The last operation result is negative (i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = Z ***Zero.***

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.

1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = C ***Carry/borrow.***

This bit is set and cleared by hardware and software. Informs if an overflow or underflow occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the “bit test and branch”, shift and rotate instructions.

STACK POINTER (SP)

Read/Write

Reset Value: 01 FFh

15							8
0	0	0	0	0	0	0	1

7							0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register always pointing to the next free location in the stack. The pointer value increments when data is taken from the stack, it decrements once data is transferred into the stack (see Figure 6).

Since the stack is 256 bytes deep, the most significant byte is forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack highest address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around the stack upper limit, without indicating a stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. You can directly manipulate the stack using PUSH and POP instructions. In case of interrupt, the PCL is stored at the first location pointed to by the SP. Other registers are then stored in the next locations as shown in Figure 6.

When interrupt is received, the SP value decrements and the context is pushed to the stack.

On return from interrupt, the SP value increments and the context is popped from the stack.

A subroutine call and interrupt occupy two and five locations in the stack area respectively.

Figure 6: Stack Manipulation Example

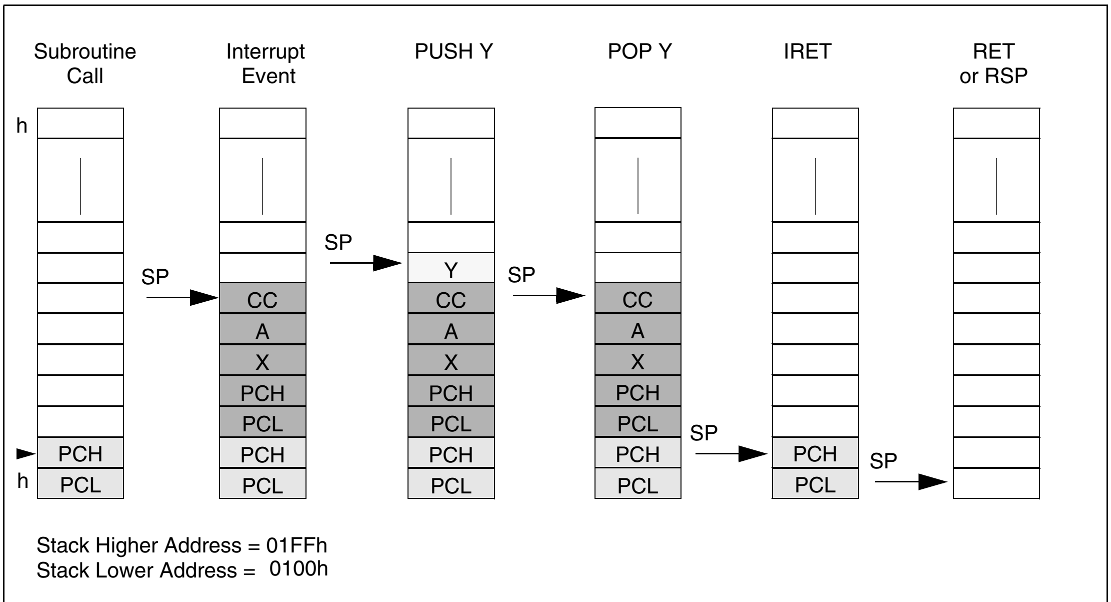


Table 4: Instruction Set (Sheet 1 of 2)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			

Table 4: Instruction Set (Sheet 2 of 2)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRXX							
Interruption management	TRAP	WFI	HALT	IRET				
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Table 5: Instruction Groups (Sheet 1 of 3)

Mnemo	Description	Function/Example	DST	SRC	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A x M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A x M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt	reset when WDG active				0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							

Table 5: Instruction Groups (Sheet 2 of 3)

Mnemo	Description	Function/Example	DST	SRC	H	I	N	Z	C
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jump if unsigned > =							
JRUGT	Jump if (C + Z = 0)	Unsigned >							
JRULE	Jump if (C + Z = 1)	Unsigned < =							
LD	Load	DST < = SRC	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	Pop reg	reg	M					
		Pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	Push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C < = DST < = C	reg, M				N	Z	C
RRC	Rotate right true C	C = > DST = > C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C < = DST < = 0	reg, M				N	Z	C
SLL	Shift left Logic	C < = DST < = 0	reg, M				N	Z	C
SRL	Shift right Logic	0 = > DST = > C	reg, M				0	Z	C
SRA	Shift right Arithmetic	DST7 = > DST = > C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C

Table 5: Instruction Groups (Sheet 3 of 3)

Mnemo	Description	Function/Example	DST	SRC	H	I	N	Z	C
SWAP	SWAP nibbles	DST[7..4] < = > DST[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	TNZ LBL1					N	Z	
TRAP	Software trap	Software interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

3 Reset

The Reset procedure provides an orderly software start-up or is used to exit Low Power modes.

Three reset modes are provided:

1. Low Voltage Detector reset,
2. Watchdog or Illegal Opcode Access reset,
3. External Reset using the $\overline{\text{RESET}}$ pin.

At reset, the reset vector is fetched from addresses FFFEh and FFFFh and loaded into the PC (the program is executed starting at this point).

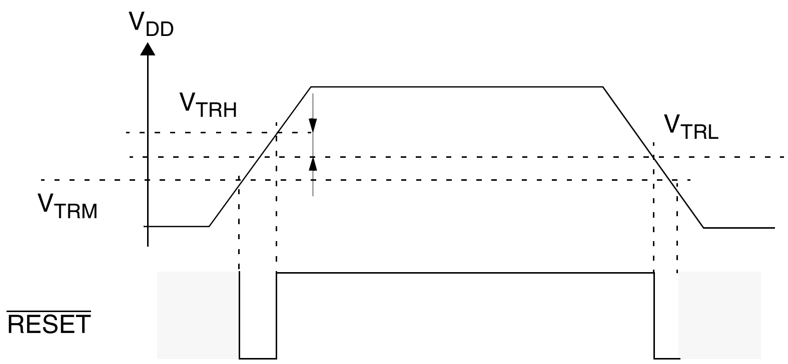
Internal circuitry provides a 4096 CPU clock cycle delay as soon as the oscillator becomes active.

3.1 Low Voltage Detector and Watchdog Reset

The Low Voltage Detector generates a reset when:

- V_{DD} is above V_{TRM},
- V_{DD} is below V_{TRH} when V_{DD} is rising,
- V_{DD} is below V_{TRL} when V_{DD} is falling (Figure 7)

Figure 7: Low Voltage Detector



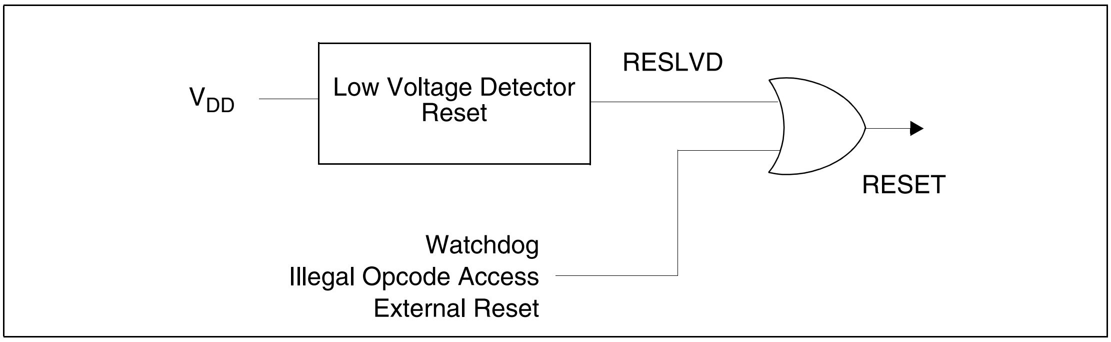
Note: Typical hysteresis (*V_{TRH}-V_{TRL}) of 50 mV.*

This circuitry is active only when V_{DD} is higher than V_{TRM}.

During the Low Voltage Detector reset, the $\overline{\text{RESET}}$ pin is held low, permitting the MCU to reset other devices.

During a Watchdog reset, the $\overline{\text{RESET}}$ pin is pulled low permitting the MCU to reset other devices as during a Low Voltage reset (Figure 8). The reset cycle is pulled low for 500 ns (typical).

Figure 8: Reset Generation Diagram



3.2 Watchdog or Illegal Opcode Access Reset

For more information about the Watchdog, please refer to Section 12: Watchdog Timer (WDG)

An Illegal Opcode reset occurs if the MCU attempts to execute a code that does not match a valid ST7 instruction.

3.3 External Reset

The external reset is an active low input signal applied to the $\overline{\text{RESET}}$ pin of the MCU.

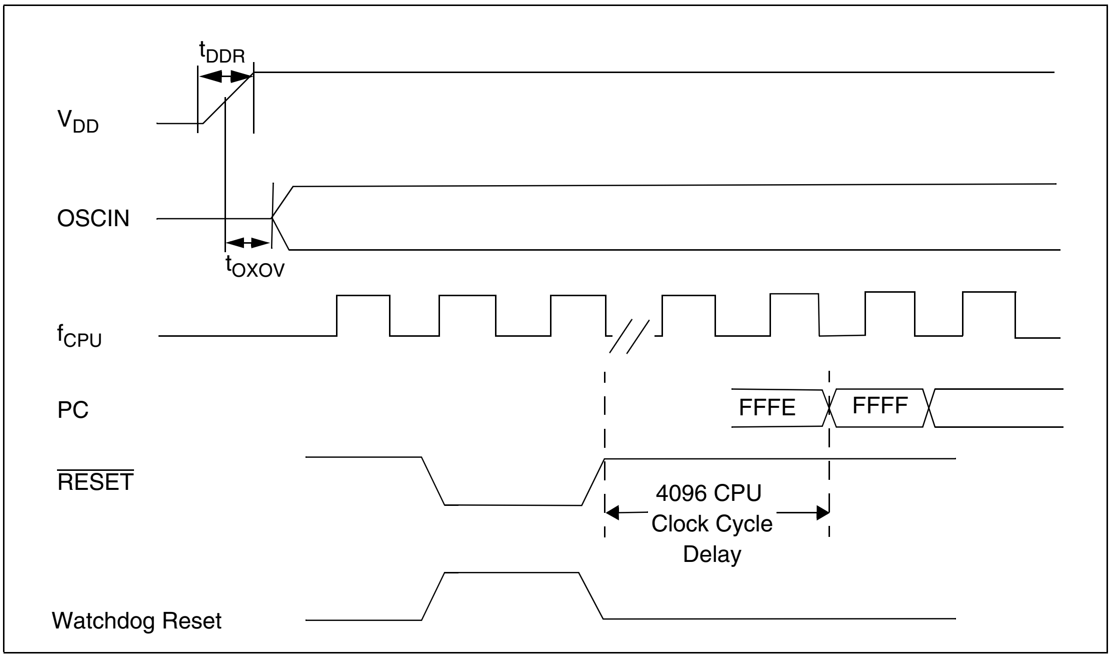
As shown in Figure 9, the $\overline{\text{RESET}}$ signal must remain low for a minimum of 1 µs.

An internal Schmitt trigger and filter provided at the $\overline{\text{RESET}}$ pin improve noise immunity.

3.4 Reset Procedure

At power-up, the MCU follows the sequence described in Figure 9.

Figure 9: Reset Timing Diagram



Note: Refer to Electrical Characteristics for values of t_{DDR}, t_{OXOV}, V_{TRH}, V_{TRL} and V_{TRM}.

4 Interrupts

There are two different methods to interrupt the ST7:

1. a maskable hardware interrupt as listed in Table 7
2. a non-maskable software interrupt (TRAP).

The Interrupt Processing flowchart is shown in Figure 10.

Only enabled maskable interrupts are serviced. However, disabled interrupts are latched and processed. For an interrupt to be serviced, the PC, X, A and CC registers are saved onto the stack, the interrupt mask (bit I of the Condition Code Register) is set to prevent additional interrupts. The Y register is not automatically saved.

The PC is then loaded with the interrupt vector and the interrupt service routine runs (refer to Table 7 for vector addresses) and ends with the IRET instruction. At the IRET instruction, the contents of the registers are recovered from the stack and normal processing resumes. Note that the I bit is then cleared if the corresponding bit stored in the stack is zero.

Though many interrupts can be run simultaneously, an order of priority is defined (see Table 7). The $\overline{\text{RESET}}$ pin has the highest priority. If the I bit is set, only the TRAP interrupt is enabled. All interrupts allow the processor to exit the WAIT Low Power mode.

4.1 Software

The software interrupt is the executable TRAP instruction. The interrupt is recognized when the TRAP instruction is executed, regardless of the state of the I bit. When an interrupt is recognized, it is serviced according to flowchart described in Figure 10.

Note: During ICC communication, the TRAP interrupt is reserved.

4.2 External Interrupts (ITA, ITB)

The ITA (PA6), ITB (PA7) pins generate an interrupt when a falling or rising edge occurs on these pins. These interrupts are enabled by the ITAITE and ITBITE bits (respectively) in the ITRFRE register, provided that the I bit from the CC register is reset. Each external interrupt has its own interrupt vector.

4.3 Peripheral Interrupts

The various peripheral devices with interrupts include both Display Data Channels (DDC A and DDC B), the Infrared Controller (IFR), two 8-bit timers (Timer A and Timer B) and the I^{2}C interface.

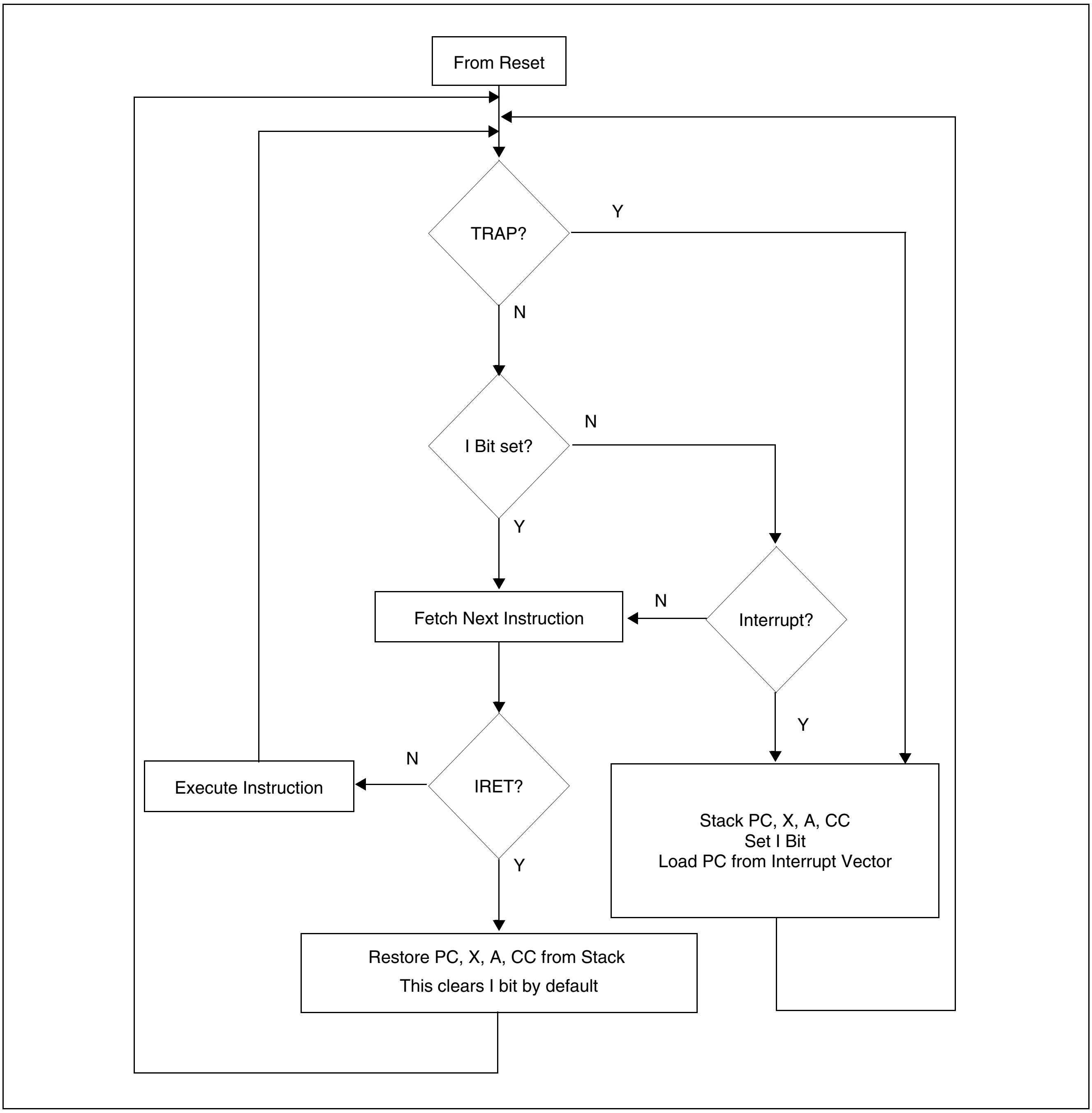
Different peripheral interrupt flags fetch an interrupt if the I bit from the CC register is reset and the corresponding Enable bit is set. If any of these conditions is not fulfilled, the interrupt is latched but not serviced, thus remaining pending.

4.4 Processing

Interrupt flags are located in the status register. The Enable bits are in the control register. When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then serviced according to the flowchart shown in Figure 10.

The general sequence for clearing an interrupt is an access to the status register when the flag is set followed by a read or write of the associated register. Note that the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting to be enabled) will therefore be lost if the Clear sequence is executed.

Figure 10: Interrupt Processing Flowchart



4.5 Register Description

Table 6: External Interrupt Register Map

Address	Reset		Register	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
000Ch	00h	R/W	ITRFRE	0	0	ITB EDGE	ITBLAT	ITBITE	ITA EDGE	ITALAT	ITAITE

EXTERNAL INTERRUPT REGISTER (ITRFRE)

Read/Write
Reset value:00h

7	6	5	4	3	2	1	0
0	0	ITBEDGE	ITBLAT	ITBITE	ITAEDGE	ITALAT	ITAITE

Bits [7:6] = Reserved. Forced by hardware to 0**.**

Bit 5 = ITBEDGE *Interrupt B Edge Selection.*
This bit is set and cleared by software.

0 Falling edge selected on ITB (default)

1 Rising edge selected on ITB

Bit 4 = ITBLAT *Falling or Rising Edge Detector Latch.*
This bit is set by hardware, when a falling or rising edge, depending on the sensitivity, occurs on the ITB/PA7 pin. An interrupt is generated if ITBITE = 1. It must be cleared by software.

0 No edge detected on ITB (default)

1 Edge detected on ITB

Bit 3 = ITBITE *ITB Interrupt Enable*.
This bit is set and cleared by software.

0 ITB interrupt disabled (default)

1 ITB interrupt enabled

Bit 2 = ITAEDGE *Interrupt A Edge Selection.*
This bit is set and cleared by software.

0 Falling edge selected on ITA (default)

1 Rising edge selected on ITA

Bit 1 = ITALAT *Falling or Rising Edge Detector Latch.*
This bit is set by hardware when a falling or a rising edge, depending on the sensitivity, occurs on the ITA/PA6 pin. An interrupt is generated if ITAITE = 1. It must be cleared by software.

0 No edge detected on ITA (default)

1 Edge detected on ITA

Bit 0 = ITAITE *ITA Interrupt Enable*.
This bit is set and cleared by software.

0 ITA interrupt disabled (default)

1 ITA interrupt enabled

Table 7: Interrupt Mapping

Source Block	Description	Register	Flag	Maskable	Vector Address	Priority Order
RESET	Reset	N/A	N/A	No	FFFEh to FFFFh	
TRAP	Software	N/A	N/A	No	FFFCh to FFFDh	Highest Priority
Not used					FFFAh to FFFBh	
DDC/CI A	DDC Interrupt	DDCSR1A DDCSR2A	**	Yes	FFF8h to FFF9h	
DDC2B A	End of communication Interrupt	DDCDCRA	ENDCF	Yes	FFF6h to FFF7h	
	End of download Interrupt		EDF	Yes	FFF6h to FFF7h	
DDC/CI B	DDC Interrupt	DDCSR1B DDCSR2B	**	Yes	FFF4h to FFF5h	
DDC2B B	End of communication Interrupt	DDCDCRB	ENDCF	Yes	FFF2h to FFF3h	
	End of download Interrupt		EDF	Yes	FFF2h to FFF3h	
Not used					FFF0h to FFF1h	
IFR	IFR Interrupt	IFRCR		Yes	FFEEh to FFEFh	
Port A bit 6	External Interrupt ITA	ITRFRE	ITALAT		FFECh to FFEDh	Lowest Priority
Port A bit 7	External Interrupt ITB	ITRFRE	ITBLAT	Yes	FFEAh to FFEBh	
I^{2}C	I^{2}C Peripheral Interrupts	I2CSR1 I2CSR2	**	Yes	FFE8h to FFE9h	
Not used					FFE6h to FFE7h	
TIMB	Timer B overflow	TIMCSRB	TOF	Yes	FFE4h to FFE5h	
TIMA	Timer A overflow	TIMCSRA	TOF	Yes	FFE2h to FFE3h	
Not used					FFE0h to FFE1h	

** Many flags can cause an interrupt, see peripheral interrupt status register description.

5 Flash Program Memory

5.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a byte-by-byte basis using an external Vpp supply.

HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using In-Circuit Programming (ICP) and In-Application Programming (IAP).

The array matrix organization allows each sector to be erased and reprogrammed without affecting other sectors.

5.2 Main Features

- Three Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (In-Circuit Programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (In-Application programming). In this mode, all sectors except Sector 0 can be programmed or erased without removing the device from the application board and when the application is running.
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Read-out protection against piracy
- Register Access Security System (RASS) to prevent accidental programming or erasing.

5.3 Structure

The Flash memory is organized in sectors and can be used for both code and data storage.

Depending on the overall size of the Flash memory in the microcontroller device, three user sectors are available. Each sector is independently erasable. Thus, having to completely erase the entire Flash memory is not necessary when only partial erasing is required.

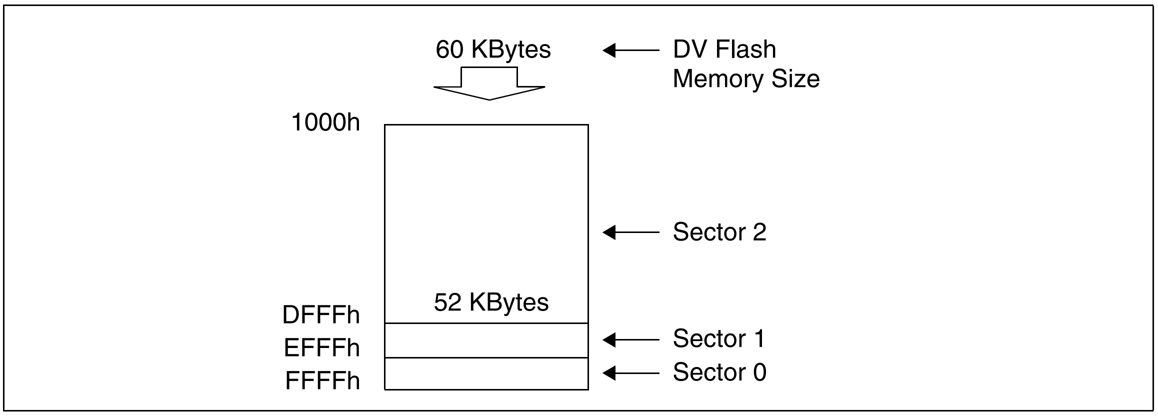
The first two sectors have a fixed size of 4 Kbytes (see Figure 11). They are mapped in the upper part of the ST7 addressing space. The reset and interrupt vectors are located in Sector 0 (F000h to FFFFh).

5.4 Program Memory Read-out Protection

The read-out protection is enabled through an option bit.

When this option is selected, the programs and data stored in the program memory (Flash or ROM) are protected against read-out piracy (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire program memory is first automatically erased. Refer to the Section 5.8 for more details.

Figure 11: Memory Map and Sector Address



5.5 In-Circuit Programming (ICP)

To perform In-Circuit Programming (ICP), the microcontroller must be switched to ICC (In-Circuit Communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations or selection of serial communication interface for downloading).

When using a STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user only needs to implement the ICP hardware interface on the application board (see Figure 12). For more details on the pin locations, refer to the device pin description.

ICP needs between 4 and 6 pins to be connected to the programming tool. Depending on the desired type of programming, these pins are:

- $\overline{\text{RESET}}$: device reset
- VSS: device power supply ground
- ICC_CLK: ICC output serial clock pin
- ICC_DATA: ICC input serial data pin
- VPP: programming voltage
- VDD: application board power supply

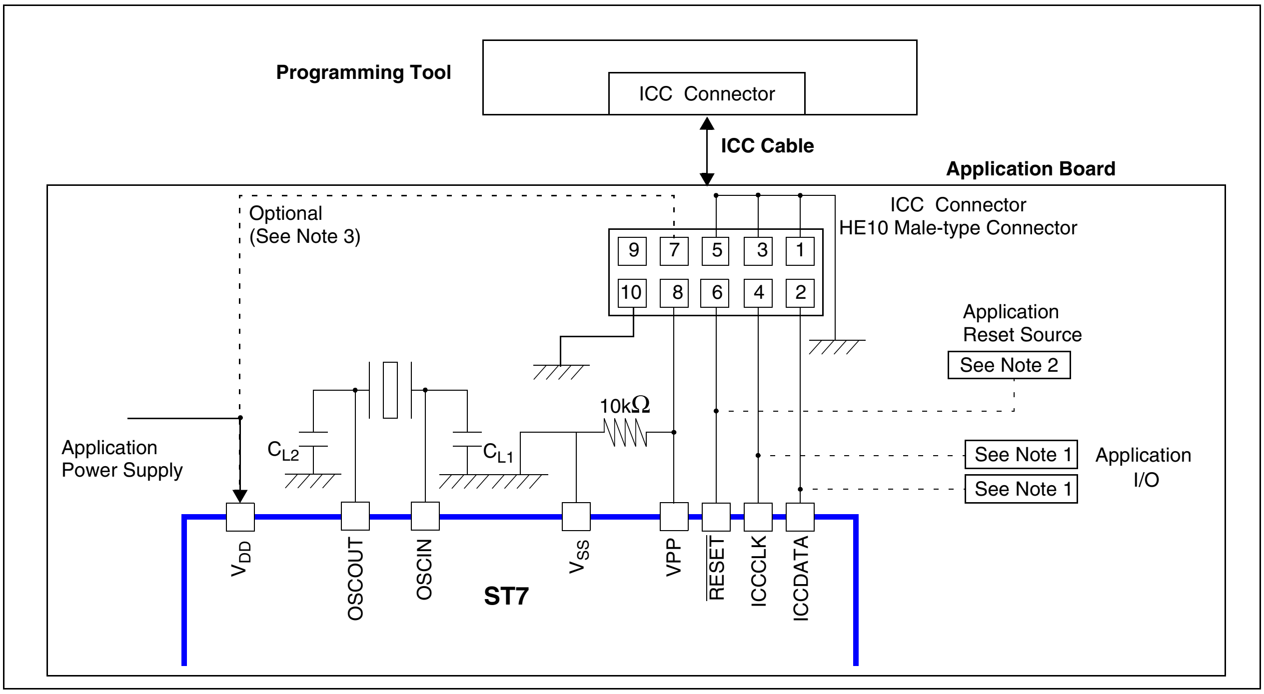
CAUTION:

1. If the ICC_CLK or ICC_DATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICC_CLK and ICC_DATA pins are not available for the application. If they are used as inputs by the application, an isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.
2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push-pull output or pull-up resistor (< 1 kΩ)). A Schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with a resistor (> 1 kΩ) or a reset management IC with open-drain output and pull-up resistor

(> 1 kΩ) , no additional components are needed. In any case, the user must ensure that an external reset is not generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

Figure 12: Typical ICP Interface



5.6 In-Application Programming (IAP)

This mode uses a Boot Loader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully-controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from either DDC interface and program it in the Flash memory. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

5.7 Register Description

FLASH CONTROL/STATUS REGISTER (FCSR)

Read/Write
Reset Value: 0000 0000 (00h)

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

This register is reserved for use by Programming Tool software. It controls the Flash programming and erasing operations.

For details on customizing Flash programming methods and In-Circuit Testing, refer to the ST7 Flash Programming Reference Manual and relevant Application Notes.

5.8 Flash Option Bytes

Each device is available for production in user programmable versions (Flash) as well as in factory coded versions (ROM). Flash devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that Flash devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

The option bytes are used to select the hardware configuration of the microcontroller. They have no address in the memory map and can be accessed only in programming mode (for example, using a standard ST7 programming tool). The default content of the Flash is fixed to FFh. To program directly the Flash devices using ICP, Flash devices are shipped to customers with the internal RC clock source enabled. In masked ROM devices, the option bytes are fixed in hardware by the ROM code.

Static Option Byte 1

	7	6	5	4	3	2	1	0
								FMP_R
Default	1	1	1	1	1	1	1	1

OPT0 = **FMP_R** *Flash memory read-out protection*
This option indicates if the user Flash memory is protected against read-out piracy. This protection is based on a read and write protection of the memory in Test and ICP modes. Erasing the option bytes when the FMP_R option is selected causes the entire user memory to be erased first.

0 Read-out protection enabled

1 Read-out protection disabled

Static Option Byte 2

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

6 Clocks & Low Power Modes

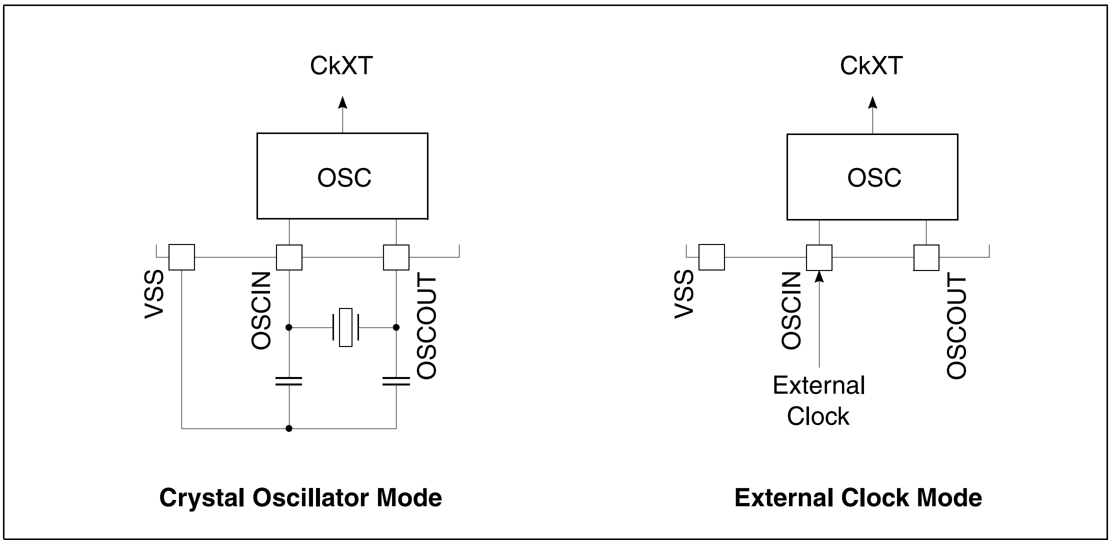
6.1 Clock System

6.1.1 General Description

The device requires a certain number of clock signals in order to operate. All clock signals are derived from the root clock signal CkXT provided at the output of the "OSC" circuit (refer to Figure 13). If a crystal oscillator or ceramic resonator is applied on pins OSCIN and OSCOUT, the OSC operates in a crystal-controlled oscillator mode. An external clock signal can also be applied on the OSCIN pin, putting the OSC in external clock mode operation.

The block diagram in Figure 13 shows the basic configuration of the clock system.

Figure 13: Main Clock Generation



6.1.2 Crystal Oscillator Mode

In this mode, the root clock is generated by the on-chip oscillator controlled by an external parallel fundamental-mode crystal oscillator or a ceramic resonator. General design precautions must be followed to ensure maximum stability. Foot capacitors C_{L1} and C_{L2} must be adapted to match the crystal oscillator or ceramic resonator. A 100-kΩ resistor is internally connected between pins OSCIN and OSCOUT.

Note: If a Murata ceramic resonator is to be used, Murata recommends their CERALOCK® CSTCG-series (fundamental type) with built-in CL1 and CL2 capacitors, such as:

- CSTCG24M0V51-R0 for 24-MHz external, 8-MHz internal clock operation

- CSTCG27M0V51-R0 for 27-MHz external, 9-MHz internal clock operation

No additional external capacitor is therefore needed with either model of this series.

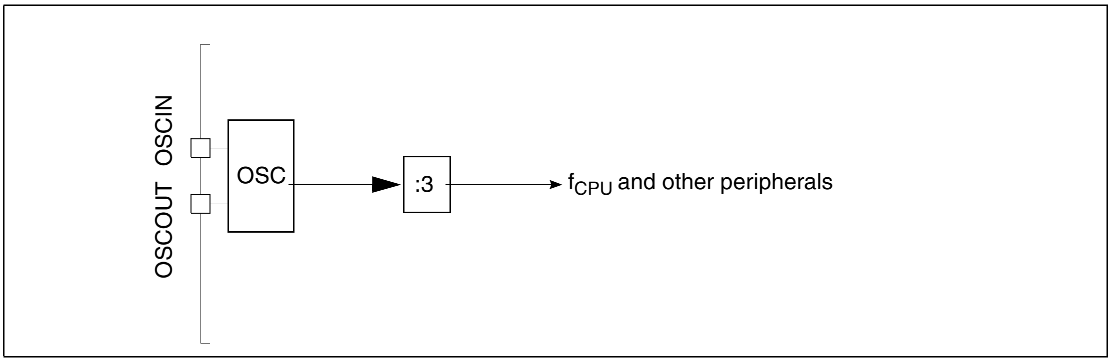
6.1.3 External Clock Mode

In this mode, an external clock is provided on pin OSCIN, while pin OSCOUT is left open. The signal is internally buffered before feeding the subsequent stages. There is the same emphasis on stability of the external clock as in Crystal Oscillator mode.

6.1.4 Clock Signals

The root clock is divided by a factor of 3 to obtain the CPU clock (f_{CPU}).

Figure 14: Clock System Diagram



6.2 Power Saving Modes

The MCU offers the possibility to decrease power consumption at any time by software operation.

6.2.1 HALT Mode

HALT mode is the MCU lowest power consumption mode. Also, HALT mode also stops the oscillator stage completely which is the most critical condition (the MCU cannot recover by itself). For this reason, HALT mode is not compatible with the watchdog protection.

Table 8: Watchdog Compatibility

Watchdog	Executing HALT Instruction
Enabled	Generates an immediate reset
Disabled	Puts the MCU in HALT mode

6.2.2 WAIT Mode

This is a low power consumption mode. The WFI instruction sets the MCU in WAIT mode. The internal clock remains active but all CPU processing is stopped. However, all other peripherals still run.

Note: In WAIT mode, DMA (DDC A and DDC B) accesses are possible.

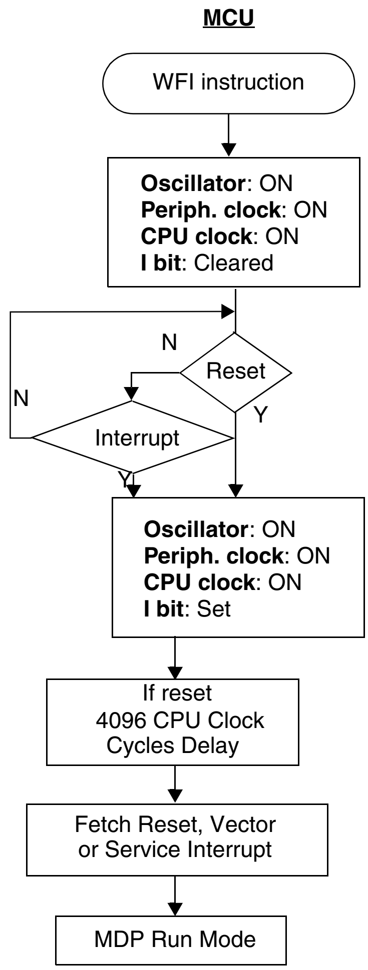
6.2.3 Exit from HALT and WAIT Modes

The MCU can exit HALT mode upon reception of an external interrupt on pins ITA or ITB. The oscillator is then turned back on and a stabilizing time is necessary before releasing CPU operation (4096 CPU clock cycles). After this delay, the CPU continues operation according to the cause of its release, either by servicing an interrupt or by fetching the reset vector in case of reset.

During WAIT mode, the I bit from the Condition Code register is cleared, enabling all interrupts. This leads the MCU to exit WAIT mode, the corresponding interrupt vector tois fetched, the interrupt routine is executed and normal processing resumes.

A reset causes the program counter to fetch the reset vector. Processing starts as with a normal reset.

Figure 15: WAIT Flow Chart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I bit is set during the interrupt routine and cleared when the CC register is popped.

6.2.4 Selected Peripherals Mode

Certain peripherals have an “On/Off “bit to disconnect the block (or part of it) and decrease MCU power consumption.

Table 9: Peripheral Modes

	Bits	Register	Comment	Default at Reset
PORTs	PxDDi	PxDDR	Cut the output function pad (input mode)	OFF
ADC	ADON	ADCSR	Cut analog consumption and clock	OFF
PWMi	OEi	PWMCRx	Cut the pad consumption	OFF
DDC	PE, DDC2BPE	DDCCR, DDCDCR		OFF
WDG	WGDA	WDGCR	Cut the output reset	OFF
I2C	PE	I2CCR		OFF

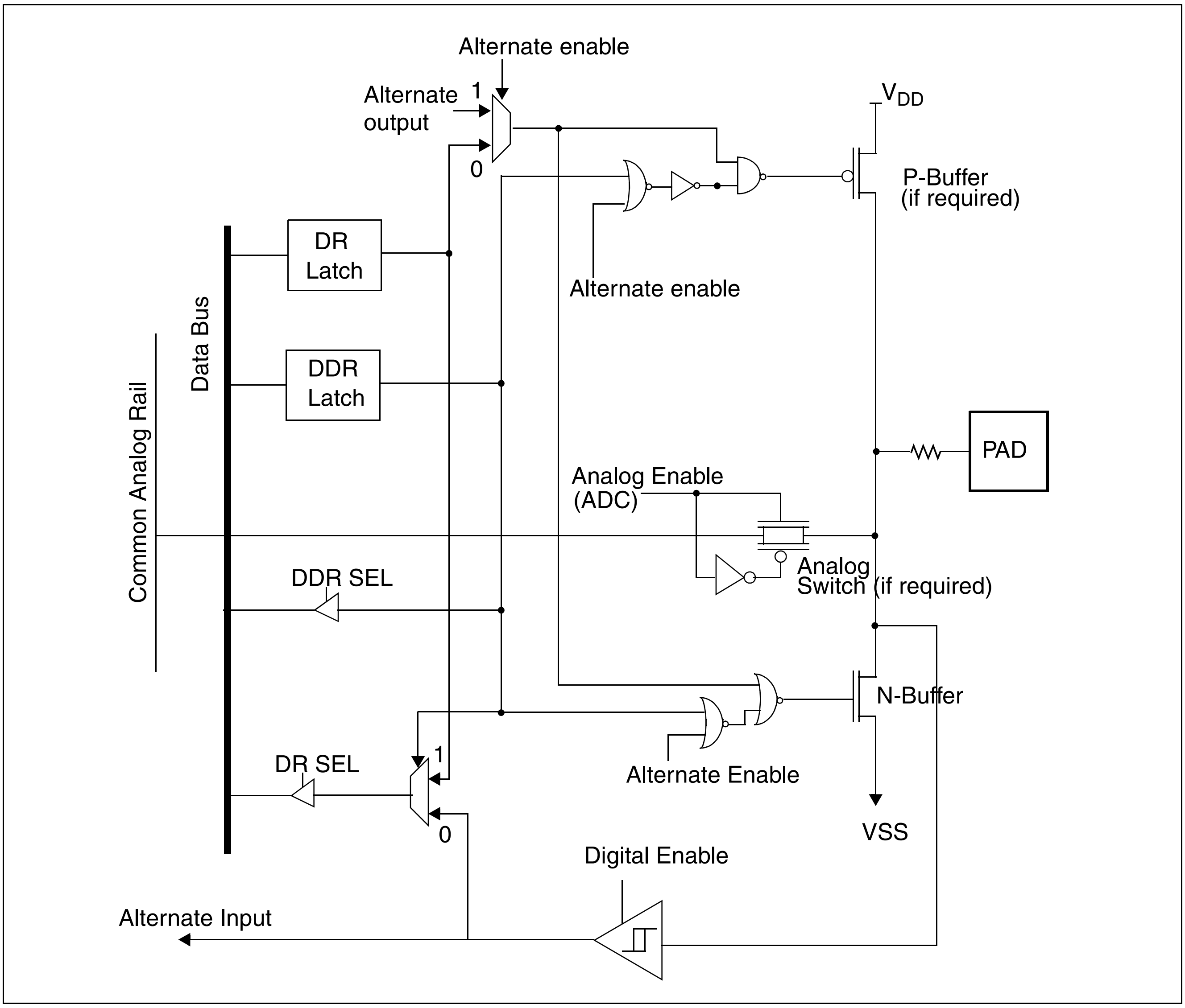
7 I/O Ports

7.1 Introduction

I/O ports are used to transfer data through digital inputs and outputs. For specific pins, I/O ports allow the input of analog signals or the Input/Output of alternate signals for on-chip peripherals (DDC, Timer, etc.).

Each pin can be independently programmed as digital input or output. Each pin can be an analog input when an analog switch is connected to the Analog-to-Digital Converter (ADC).

Figure 16: I/O Pin Critical Circuit



Note:1. This is a typical I/O pin configuration. Each port is customized with a specific configuration in order to handle certain functions.

Table 10: I/O Pin Function

DDR	Mode
0	Input
1	Output

7.2 Common Functional Description

Each port pin of the I/O Ports can be individually configured as either an input or an output, under software control.

Each bit of Data Direction Register (DDR) corresponds to an I/O pin of the associated port. This corresponding bit must be set to configure its associated pin as an output and must be cleared to configure its associated pin as an input (see Note 1 on page 35). The Data Direction Registers can be read and written.

A typical I/O circuit is shown in Figure 16. Any write to an I/O port updates the port data register even when configured as an input. Any read of an I/O port returns either the data latched in the port data register (pins configured as output) or the value of the I/O pins (pins configured as an input).

Remark: When there is no I/O pin inside an I/O port, the returned value is logic 0 (pin configured as an input).

At reset, all DDR registers are cleared, configuring all I/O ports as inputs. Data Registers (DR) are also cleared at reset.

Input mode

When DDR = 0, the corresponding I/O is configured in Input mode.

In this case, the output buffer is switched off and the state of the I/O is readable through the Data Register address, coming directly from the TTL Schmitt Trigger output and not from the Data Register output.

Output mode

When DDR = 1, the corresponding I/O is configured in Output mode.

In this case, the output buffer is activated according to the Data Register content.

A read operation is directly performed from the Data Register output.

Analog input

Each I/O can be used as an analog input by adding an analog switch driven by the ADC. The I/O must be configured as an input before using it as analog input.

When the analog channel is selected by the ADC, the analog value is directly driven to the ADC through an analog switch.

Alternate mode

A signal coming from an on-chip peripheral is output on the I/O which is then automatically configured in output mode.

The signal coming from the peripheral enables the alternate signal to be output. A signal coming from an I/O can be input to an on-chip peripheral.

An alternate Input must first be configured in Input mode (DDR = 0). Alternate and I/O Input configurations are identical without pull-up. The signal to be input in the peripheral is taken after the TTL Schmitt trigger when available.

The I/O state is readable as in Input mode by addressing the corresponding I/O Data Register.

7.3 Port A

Each Port A bit can be defined as an Input line or as a Push-Pull. It can be also be used to output the PWM outputs.

Table 11: Port A Description

Port A	I / O		Alternate Function	
	Input[1]	Output	Signal	Condition
PA0	with Weak Pull-up	Push-pull	PWM0	OE0 = 1 (PWM)
PA1	with Weak Pull-up	Push-pull	PWM1	OE1 = 1 (PWM)
PA2	with Weak Pull-up	Push-pull	PWM2	OE2 = 1 (PWM)
PA3	with Weak Pull-up	Push-pull	PWM3	OE3 = 1 (PWM)
PA4	with Weak Pull-up	Push-pull	PWM4	OE4 = 1 (PWM)
PA5	with Weak Pull-up	Push-pull	PWM5	OE5 = 1 (PWM)
	with Weak Pull-up		BUZOUT	BUZEN = 1 (Timer A)[2]
PA6	with Weak Pull-up	Push-pull	External Interrupt ITA	see External Interrupt Register Description
PA7	with Weak Pull-up	Push-pull	External Interrupt ITB	

1. Reset state.
2. If both PWM5 and BUZOUT are enabled, BUZOUT has priority over PWM5.

Outputs PA4 and PA5 may also be configured as high current (8 mA) push-pull outputs by means of the MISCR register.

MISCELLANEOUS REGISTER (MISCR)

Read/Write
Reset value:00h

7	6	5	4	3	2	1	0
0	0	0	0	0	PA5OVD	PA4OVD	0

Bits [7:3] = Reserved. Forced by hardware to 0**.**

Bit 2 = PA5OVD *Port A Bit 5 Overdrive*
This bit is set and cleared by software. It is used only if Port A Bit 5 is set as an output (PADDR, PWM5 or BUZOUT). It has no effect if set as an input.

0 2 mA Push-pull Output

1 8 mA Push-pull Output

Bit 1 = PA4OVD *Port A Bit 4 Overdrive*
This bit is set and cleared by software. It is used only if Port A Bit 4 is set as an output (PADDR or PWM4). It has no effect if set as an input.

0 2 mA Push-pull Output

1 8 mA Push-pull Output

Bit 0 = Reserved. Must be cleared by software.

Figure 17: Port A [5:0]

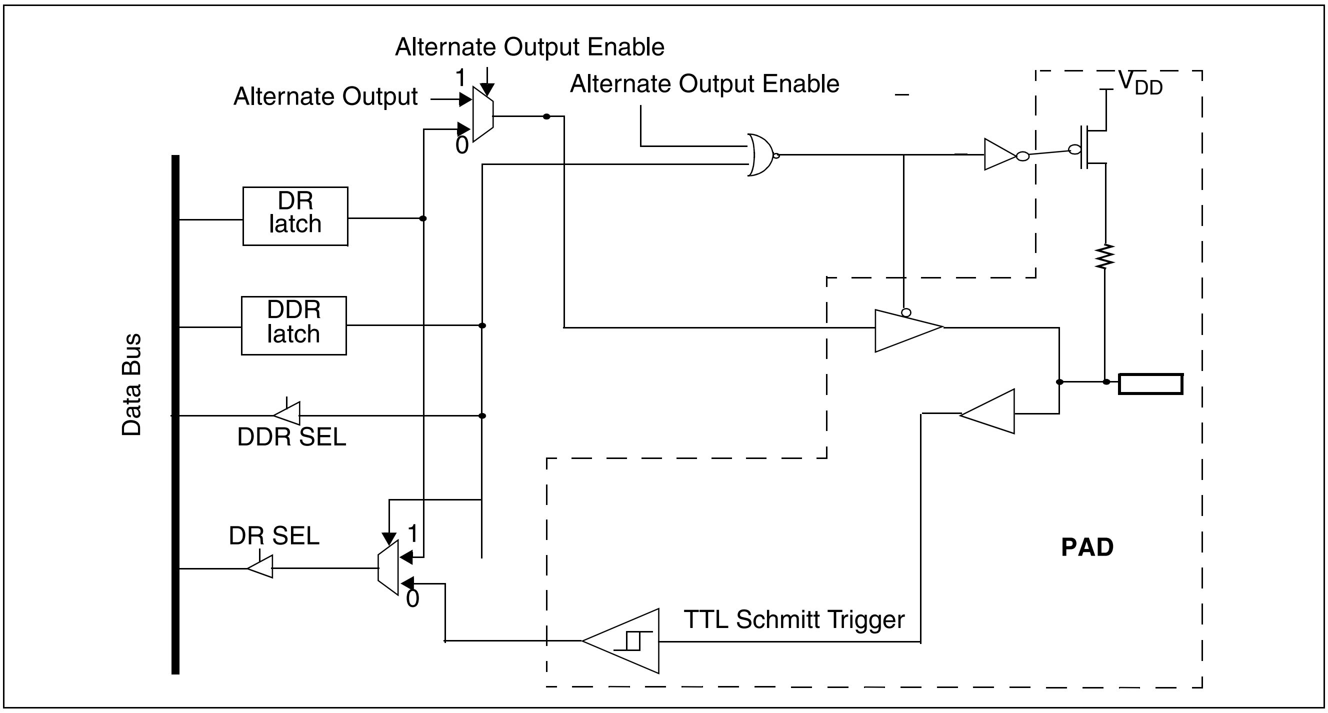
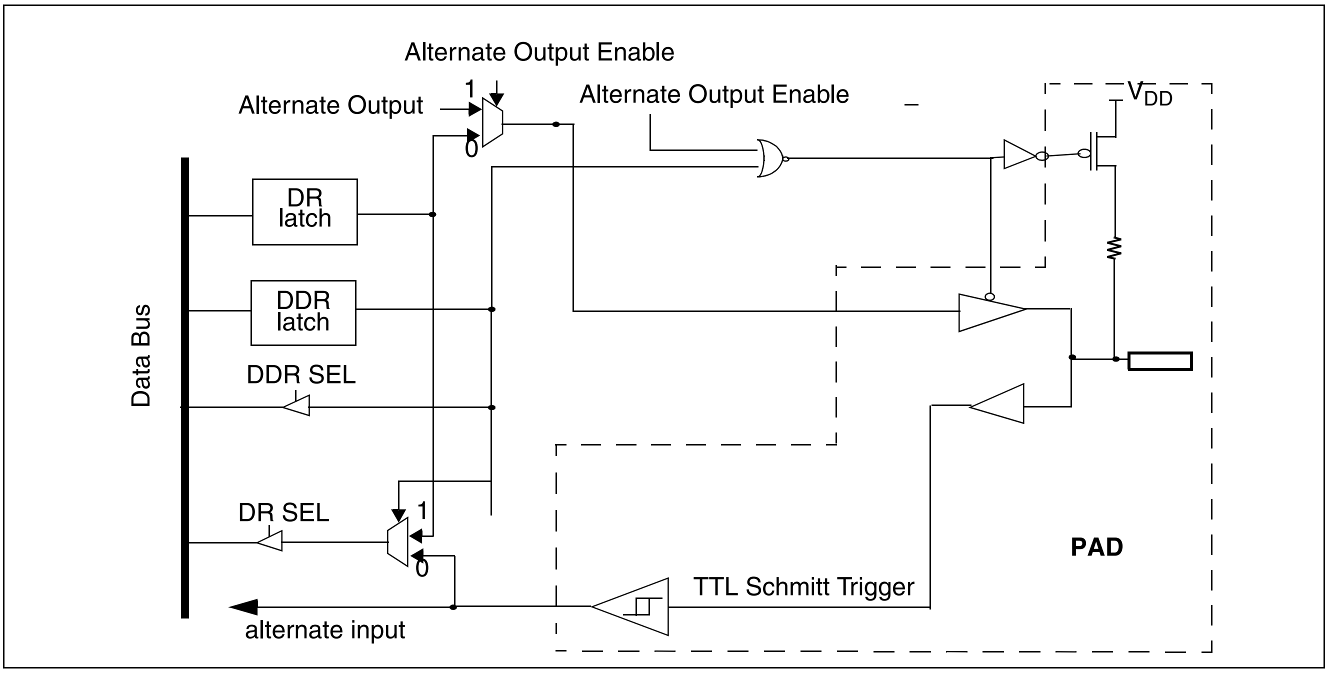


Figure 18: Port A [7:6]



7.4 Port B

Each Port B bit can be used as the Analog source to the Analog-to-Digital Converter.

Only one I/O line at a time must be configured as an analog input. Pins levels are all limited to 5V.

All unused I/O lines should be tied to an appropriate logic level (either V_{DD} or V_{SS}).

Since ADC and microprocessor are on the same chip and if high precision is required, the user should not switch heavily loaded signals during conversion. Such switching will affect the supply voltages used as analog references. The conversion accuracy depends on the quality of power supplies (V_{DD} and V_{SS}). The user must take special care to ensure that a well regulated reference voltage is present on pins V_{DD} and V_{SS} (power supply variations must be less than 3.3 V/ms). This implies, in particular, that a suitable decoupling capacitor is used at pin V_{DD}.

Table 12: Port B Description

PORT B	I / O		Alternate Function	
	Input[1]	Output	Signal	Condition
PB0	with Weak Pull-up when Digital Input	Push-pull	Analog Input (ADC):AIN0	ADON = 1 & CH[1:0] = 00 (ADCCSR)
PB1	with Weak Pull-up when Digital Input	Push-pull	Analog Input (ADC) AIN1	ADON = 1 & CH[1:0] = 01 (ADCCSR)
PB2	with Weak Pull-up when Digital Input	Push-pull	Analog Input (ADC) AIN2	ADON = 1 & CH[1:0] = 10 (ADCCSR)
PB3	with Weak Pull-up when Digital Input	Push-pull	Analog Input (ADC) AIN3/ IFR	ADON = 1 & CH[1:0] = 11 (ADCCSR) for analog input. In this case, IFR is disabled.

1. Reset state.

Figure 19: Port B [2:0]

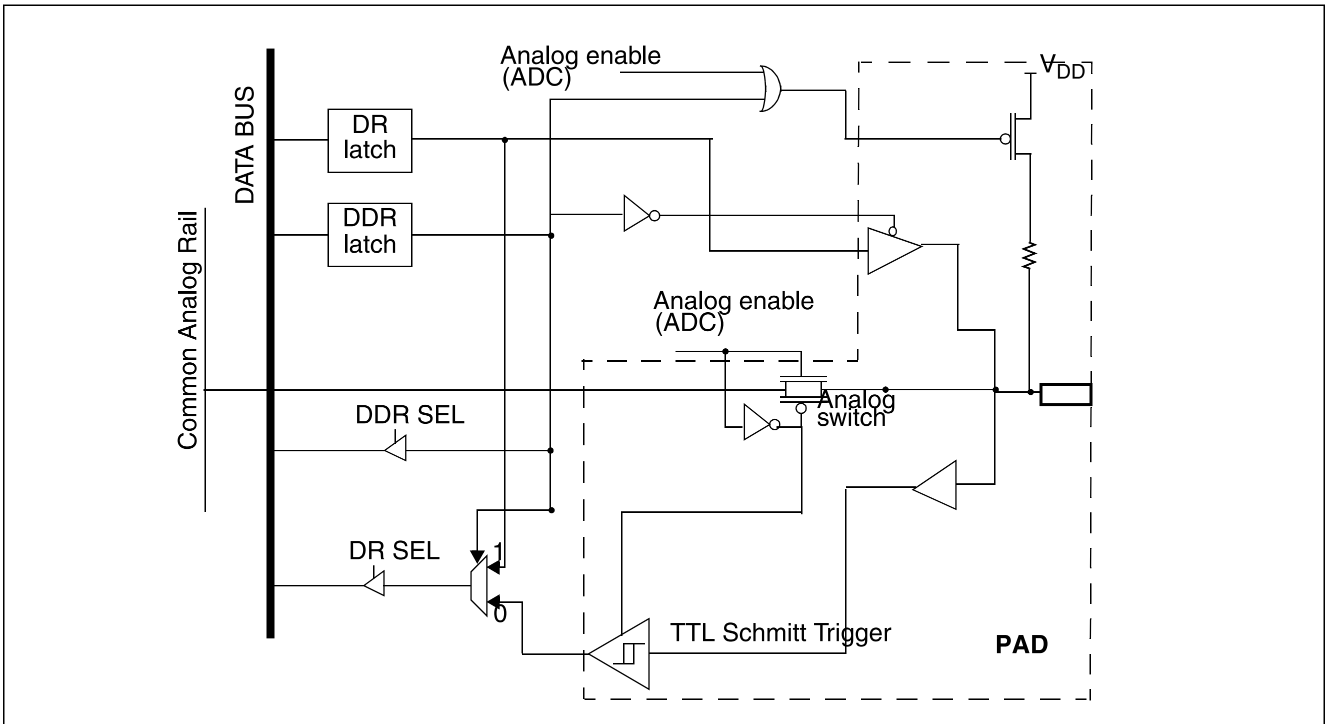
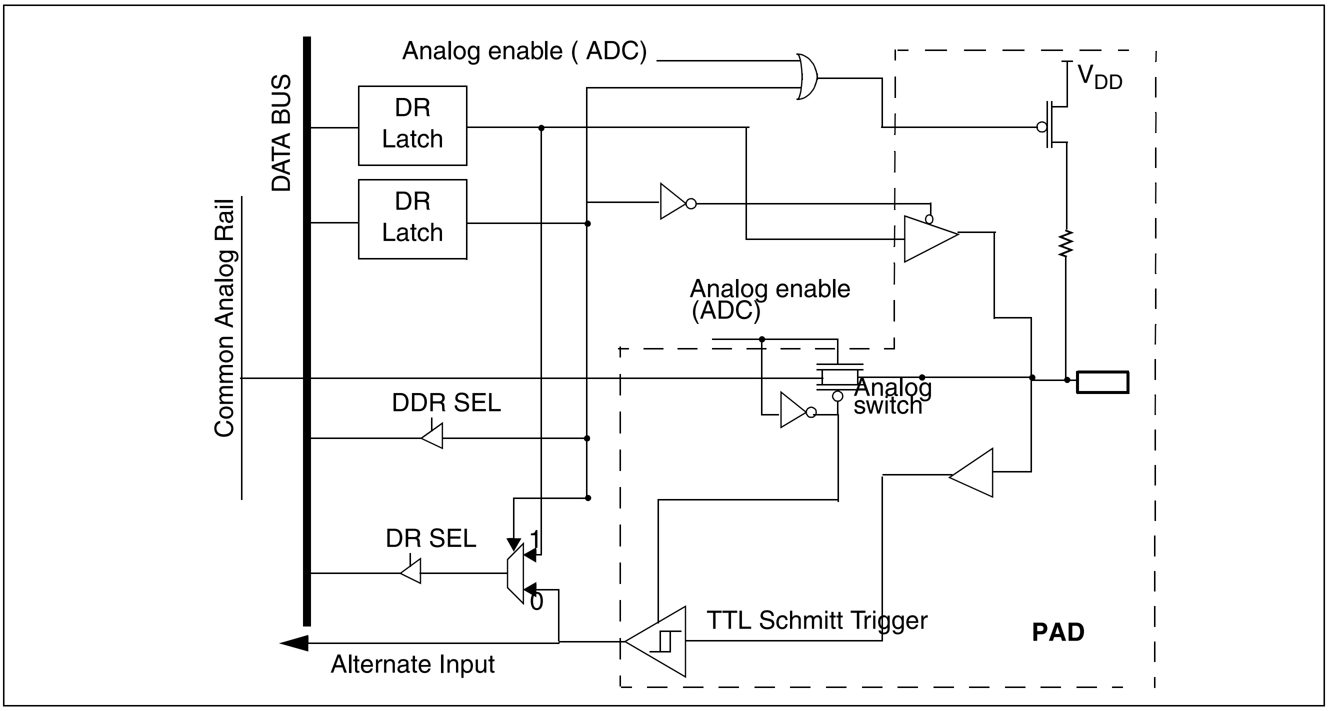


Figure 20: Port B [3]



7.5 Port C

The available port pins of port C may be used as general purpose I/Os.

Table 13: Port C Description

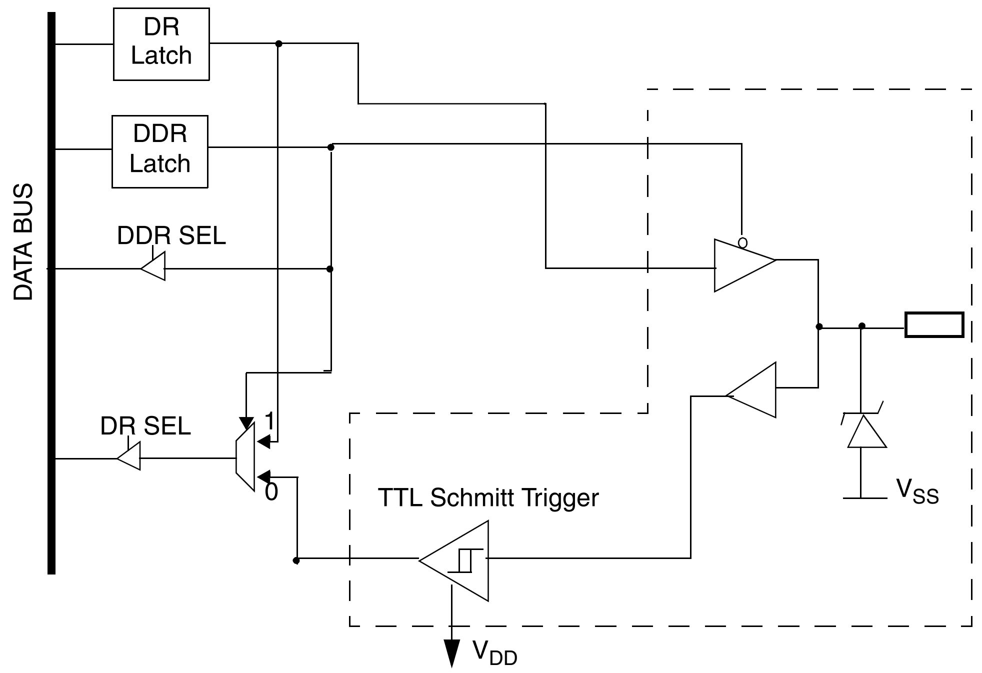
PORT C	I / O		Alternate Function	
	Input[1]	Output	Signal	Condition
PC0	Without Pull-up	Open-drain		
PC1	Without Pull-up	Open-drain		

1. Reset state.

For more information, refer to the relevant Application Notes.

Note: *These 2 pins are reserved for ICC use during ICC communication. If ICC is not used at all, they can be used as general purpose I/Os.*

Figure 21: Port C



7.6 Port D

The alternate functions are:

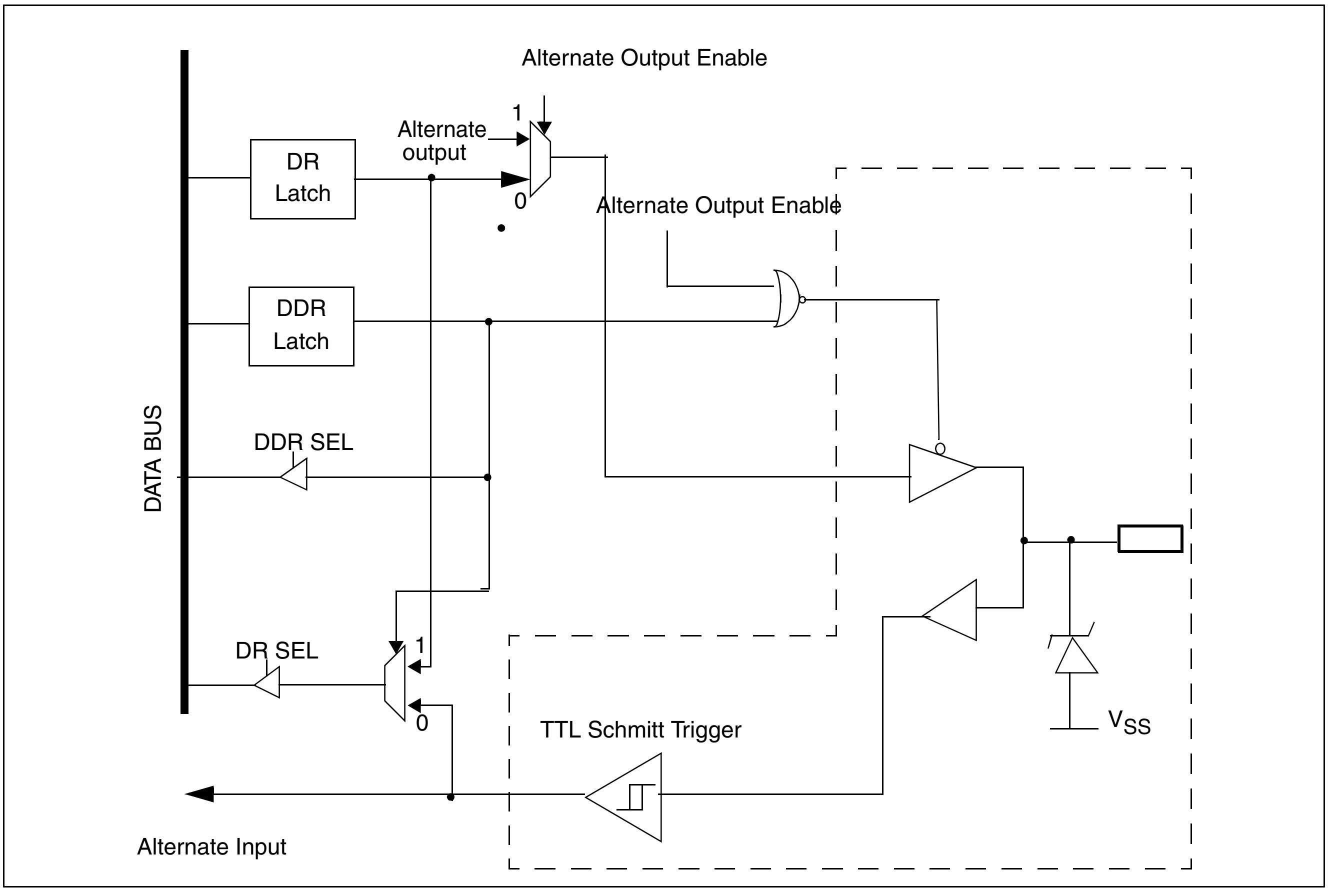
- the I/O pins of the on-chip I²C SCLI & SDAI for PD[1:0],
- the I/O pins of the on-chip DDC A SCLD & SDAD for PD[3:2],
- the I/O pins of the on-chip DDC B SCLD & SDAD for PD[5:4]
- input and output on PD[7:6].

Table 14: Port D Description

PORT D	I / O		Alternate Function	
	Input[1]	Output	Signal	Condition
PD0	Without Pull-up	Open-drain	SCLI (input with TTL Schmitt trigger or Open-drain output)	I²C enable
PD1	Without Pull-up	Open-drain	SDAI (input with TTL Schmitt trigger or Open-drain output)	I²C enable
PD2	Without Pull-up	Open-drain	SCLD A (input with TTL Schmitt trigger or Open-drain output)	DDC A enable
PD3	Without Pull-up	Open-drain	SDAD A (input with TTL Schmitt trigger or Open-drain output)	DDC A enable
PD4	Without Pull-up	Open-drain	SCLD B (input with TTL Schmitt trigger or Open-drain output)	DDC B enable
PD5	Without Pull-up	Open-drain	SDAD B (input with TTL Schmitt trigger or Open-drain output)	DDC B enable
PD6	Without Pull-up	Open-drain		
PD7	Without Pull-up	Open-drain		

1. Reset state.

Figure 22: Port D



7.7 Register Description

DATA REGISTERS (PXDR)

DATA DIRECTION REGISTERS (PXDDR)

('x' corresponds to the I/O pin of the associated port. In Input mode, the value is 00h by default).I

Table 15: I/O Port Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0002h	00h	R/W	PADR	PADR[7:0]							
0003h	00h	R/W	PADDR	PADDR[7:0]							
0004h	00h	R/W	PBDR	PBDR[7:0]							
0005h	00h	R/W	PBDDR	PBDDR[7:0]							
0006h	00h	R/W	PCDR	PCDR[7:0]							
0007h	00h	R/W	PCDDR	PCDDR[7:0]							
0008h	00h	R/W	PDDR	PDDR[7:0]							
0009h	00h	R/W	PDDDR	PDDDR[7:0]							

8 PWM Generator

8.1 Introduction

This PWM on-chip peripheral consists of two blocks, each one with its own 8-bit auto-reload counter.

The first block (Block A) outputs up to 4 separate PWM signals at the same frequency. The second block (Block B) outputs up to 2 separate PWM signals at another frequency.

Each PWM output may be enabled or disabled independently of the other. The polarity of each PWM output may also be independently set.

8.2 Main Features

- 2 distinct programmable frequencies between 31.250 kHz and 8 MHz.
- Resolution: t_{CPU}

8.3 Functional Description

The free-running 8-bit counter is fed by the CPU clock and increments on every rising edge of the clock signal.

When a counter overflow occurs, the counter is automatically reloaded with the contents of the ARR register.

Each PWMx output signal can be enabled independently using the corresponding OEx bit in the PWM control register (PWMCR). When this bit is set, the corresponding I/O is configured as an output push-pull alternate function.

PWM[3:0] all have the same frequency which is controlled by counter period A and the ARRA register value.

$$f_{PWMA} = f_{COUNTERA} / (256\text{-}ARRA)$$

PWM[5:4] all have the same frequency which is controlled by counter period B and the ARRB register value.

$$f_{PWMB} = f_{COUNTERB} / (256\text{-}ARRB)$$

When a counter overflow occurs, the PWMx pin level is toggled depending on the corresponding OPx (output polarity) bit in the PWMCR register. When the counter reaches the value contained in one of the Duty Cycle registers (DCRIx), the corresponding PWMx pin level is restored.

This DCRIx register can not be accessed directly, it is loaded from the Duty Cycle register (DCRx) at each overflow of the counter. This double buffering method prevents glitch generation when changing the duty cycle on the fly.

Note that the reload values will also affect the value and the resolution of the duty cycle of the PWM output signal. To obtain a signal on a PWMx pin, the contents of the DCRx register must be greater than or equal to the contents of the ARR register. The maximum available resolution for duty cycle is 1/(256-ARR).

Figure 23: PWM Block Diagram

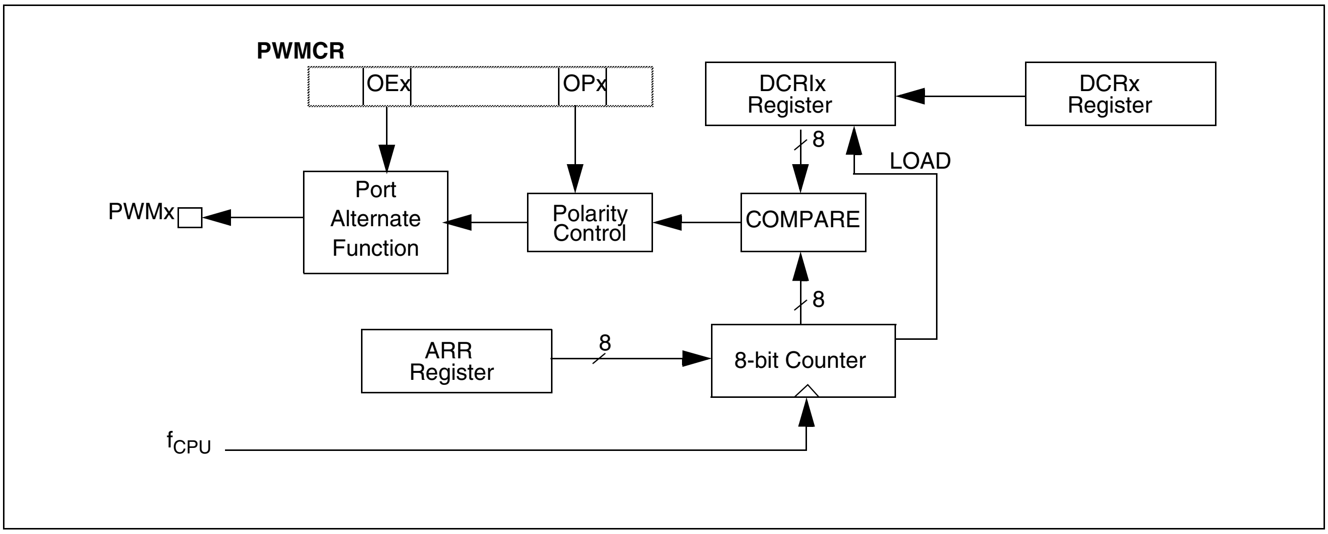


Figure 24: PWM Generation

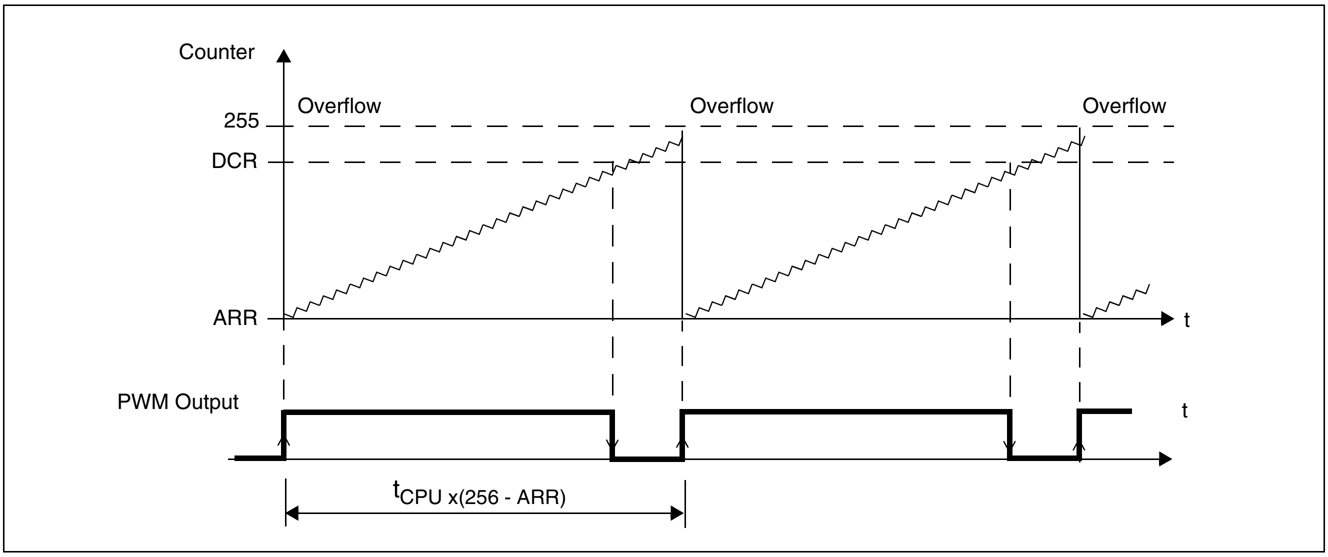
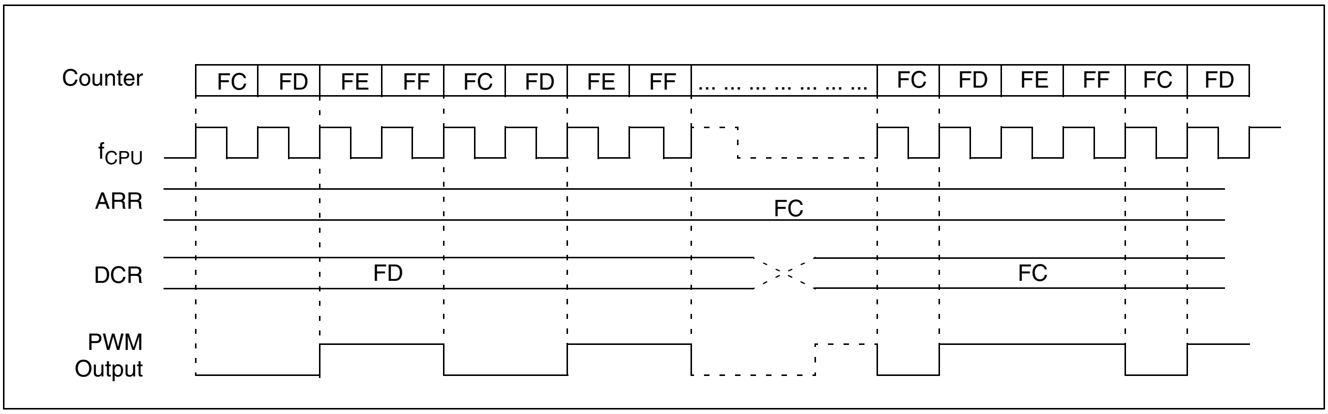


Figure 25: PWM Generation



Equations:

Table 16: Pulse WIdth in t_{CPU}

	Pulse WIdth in t_{CPU}
DCR ≥ ARR	DCR - ARR + 1
DCR = ARR	1
DCR < ARR	0 (Output will not toggle)

$$\text{Duty Cycle} = \frac{DCR + 1}{256 - ARR}$$

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 through 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.

Table 17: 8-bit PWM Ripple after Filtering

C_{EXT}	V_{RIPPLE}
470 nF	60 mV
1 µF	27 mV
4.7 µF	6 mV

$$V_{RIPPLE} = \frac{(1 - e^{1/(2 \times C_{EXT} \times R_{EXT} \times f_{PWM})})^2}{|1 - e^{1/(C_{EXT} \times R_{EXT} \times f_{PWM})}|} \times V_{DD}$$

With:

$R_{EXT} = 1\ k\Omega$

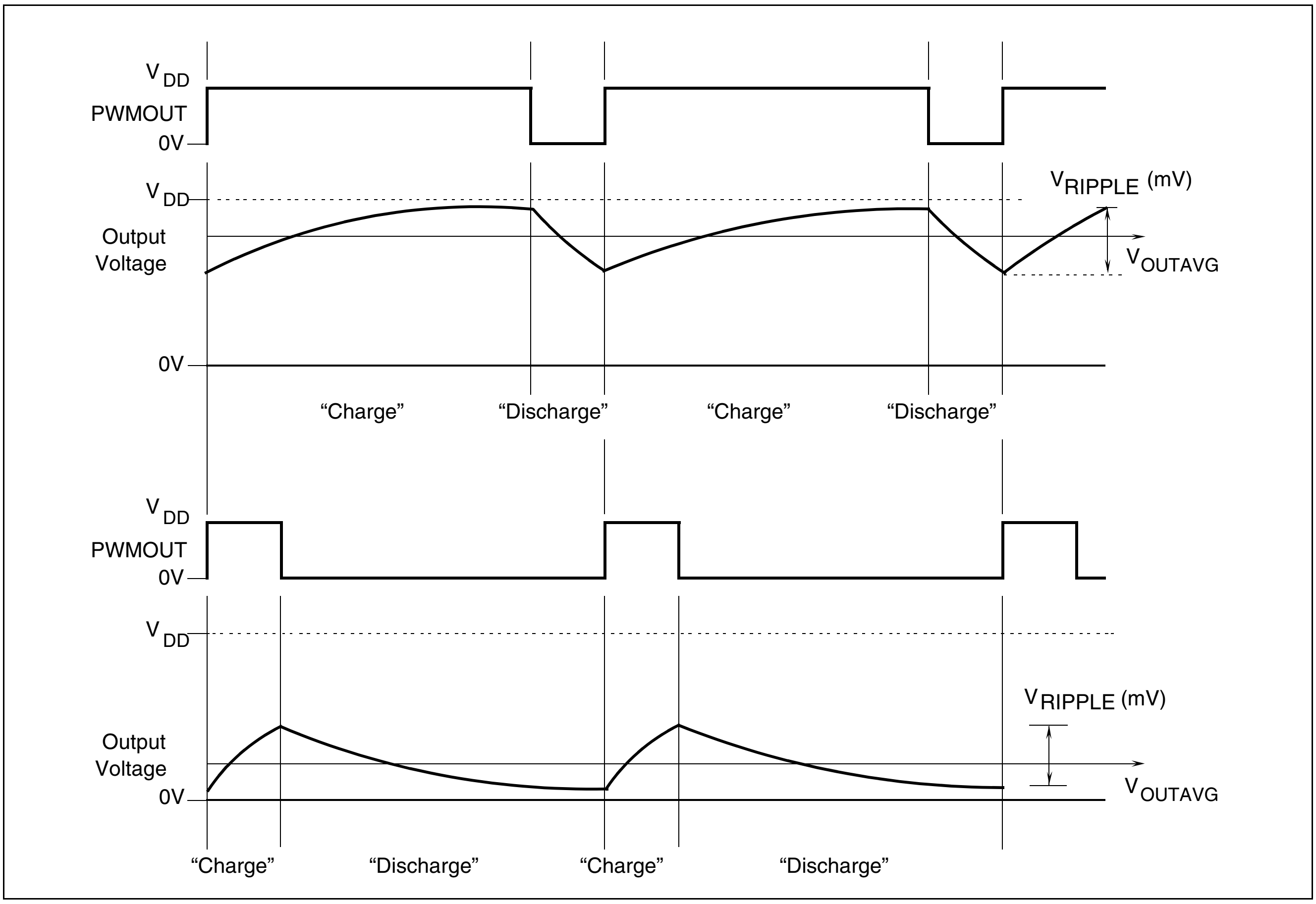
$f_{PWM} = f_{CPU} / (256 - ARR)$

$f_{CPU} = 8\ MHz$

$V_{DD} = 5\ V$

Worst case, PWM Duty Cycle 50%

Figure 26: PWM Simplified Voltage Output after Filtering



8.4 Register Description

Each PWM is associated with two control bits (OEx and OPx) and a control register (DCRx).

Table 18: PWM Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000Fh	00h	R/W	PWMDCR0	DCR0[7:0]							
0010h	00h	R/W	PWMDCR1	DCR1[7:0]							
0011h	00h	R/W	PWMDCR2	DCR2[7:0]							
0012h	00h	R/W	PWMDCR3	DCR3[7:0]							
0013h	00h	R/W	PWMCRA	OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0
0014h	FFh	R/W	PWMARRA	ARRA[7:0]							
0015h	00h	R/W	PWMDCR4	DCR4[7:0]							
0016h	00h	R/W	PWMDCR5	DCR5[7:0]							
0017h	00h	R/W	PWMCRB	0	0	OE5	OE4	0	0	OP5	OP4
0018h	FFh	R/W	PWMARRB	ARRB[7:0]							

DUTY CYCLE REGISTERS (PWMDCRx)

Read/Write
Reset Value 0000 0000 (00h)

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

Bits [7:0] = DC[7:0] *Duty Cycle Data*

These bits are set and cleared by software.
A DCRx register is associated with the DCRix register of each PWM channel to determine the second edge location of the PWM signal (the first edge location is common to all 4 channels and given by the ARR register). These DCR registers allow the duty cycle to be set independently for each PWM channel.

CONTROL REGISTER A (PWMCRA)

Read/Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0

Bits [7:4] = OE [3:0] *PWM Output Enable.*

These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin.

0 the PWM pin is a general I/O.

1 the PWM pin is driven by the PWM peripheral.

Bits [3:0] = OP[3:0] *PWM Output Polarity.*

These bits are set and cleared by software. They independently select the polarity of the 4 PWM output signals.

0 positive polarity.

1 negative polarity.

Note: When an OPx bit is modified, the PWMx output signal is immediately updated.

AUTO-RELOAD REGISTER A (PWMARRA)

Read/Write
Reset Value: 1111 1111(FFh)

7	6	5	4	3	2	1	0
AR73	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bits [7:0] = AR[7:0] *Counter Auto-Reload Data.*

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. Writing in this register reload the PWM counter to ARR A value. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register adjusts the PWM frequency (setting the PWM duty cycle resolution) for outputs PWM[3:0].

CONTROL REGISTER B (PWMCRB)

Read/Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	OE5	OE4	0	0	OP5	OP4

Bits [7:6] = Reserved. Forced by hardware to 0.

Bits [5:4] = OE[5:4] *PWM Output Enable.*

These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin.

0 the PWM pin is a general I/O.

1 the PWM pin is driven by the PWM peripheral.

Bits [3:2] = Reserved. Forced by hardware to 0.

Bit [1:0] = OP[5:4] *PWM Output Polarity.*

These bits are set and cleared by software. They independently select the polarity of the 4 PWM output signals.

0 positive polarity.

1 negative polarity.

Note: When an OPx bit is modified, the PWMx output signal is immediately reversed.

AUTO-RELOAD REGISTER B (PWMARRB)

Read/Write
Reset Value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
AR73	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bits [7:0] = AR [7:0] *Counter Auto-Reload Data.*

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. Writing in this register reload the PWM counter to ARR B value. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register adjusts the PWM frequency (by setting the PWM duty cycle resolution) for outputs PWM[5:4].

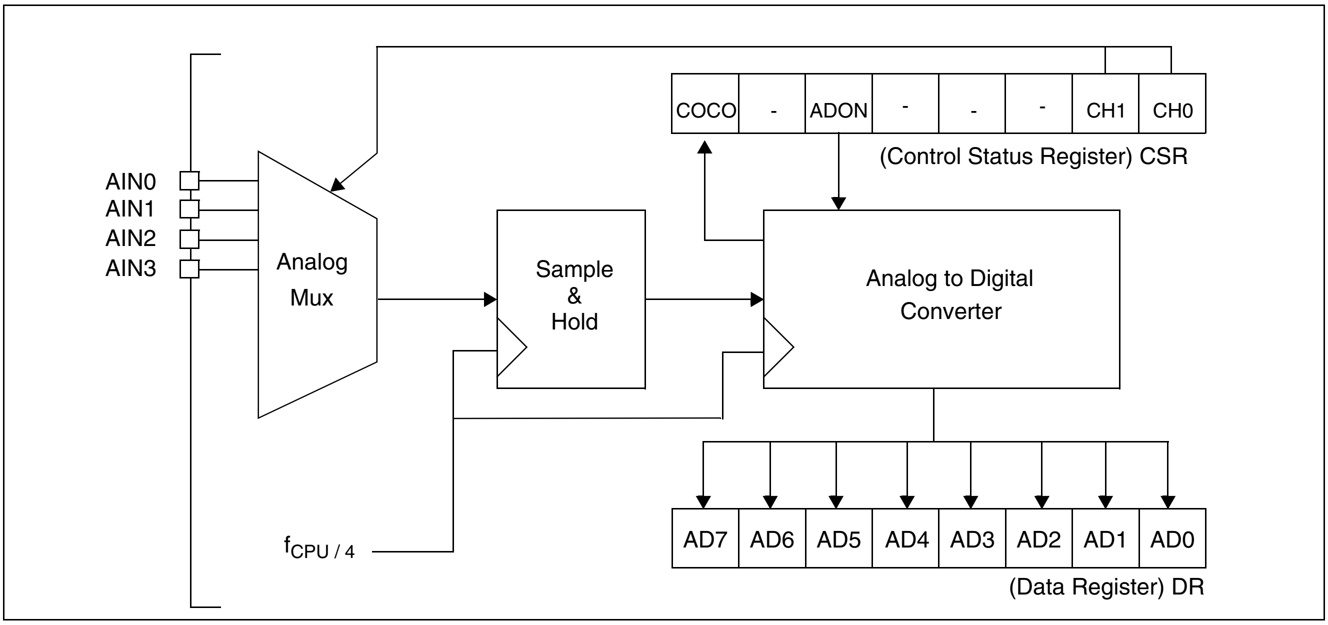
9 8-bit Analog-to-Digital Converter (ADC)

9.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter with internal Sample and Hold circuitry. This peripheral has up to 4 multiplexed analog input channels (refer to device pin out description) that allows the peripheral to convert the analog voltage levels from up to 4 different sources.

The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

Figure 27: ADC Block Diagram



9.2 Main Features

- 8-bit conversion
- Up to 4 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/Off bit (to reduce power consumption)

9.3 Functional Description

The high and low level reference voltages are V_{DD} and V_{SS}, respectively. Consequently, conversion accuracy is degraded by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

Characteristics

The conversion is monotonic, the result never decreases or increases if the analog input does not also drecrease or increase.

If the input voltage is greater than or equal to V_{DD} (voltage reference high), the results are equal to FFh (full scale) without overflow indication.

If the input voltage is less than or equal to V_{SS} (voltage reference low), the results are equal to 00h.

The A/D converter is linear, the digital result of the conversion is given by the formula:

$$\text{Digital result} = \frac{255 \text{ x Input Voltage}}{\text{Supply Voltage}}$$

The conversion accuracy is described in Section 17: Electrical Characteristics.

When the A/D converter is continuously "ON", the conversion time is 16 ADC clock cycles which corresponds to 64 CPU clock cycles.

The internal circuitry is in auto-calibration during the conversion cycle. This process prevents offset drifts. Still, calibration cycles are required at start-up or after any A/D converter re-start.

Procedure

Refer to the CSR and SR registers in Section 9.4: Register Description for the bit definitions.

At start-up, the A/D converter is OFF (ADON bit equal to '0').

Prior to using the A/D converter, the analog input ports must be configured as inputs. Refer to Section 7: I/O Ports. Using these pins as analog inputs does not affect the ability to read the port as a logic input.

Then, the ADON bit must be set to 1. As internal AD circuitry starts calibration, it is mandatory to respect the stabilizing time (several tens of milliseconds) prior to using A/D results.

In the CSR register, bits CH1 to CH0 select the analog channel to be converted (see Table 19). These bits are set and cleared by software.

The A/D converter performs a continuous conversion of the selected channel.

When a conversion is complete, the COCO bit is set by hardware, but no interrupt is generated. The result is written in the DR register.

Reading the DR result register resets the COCO bit.

Writing to the CSR register aborts the current conversion, the COCO bit is reset and a new conversion is started.

Note: *Resetting the ADON bit disables the A/D converter. Thus, power consumption is reduced when no conversions are needed.*

The A/D converter is not affected by WAIT mode.

9.4 Register Description

Table 19: ADC Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000Ah	00h	R	ADCDR	AD[7:0]							
000Bh	00h	R/W	ADCCSR	COCO	0	ADON	0	0	0	CH[1:0]	

CONTROL/STATUS REGISTER (ADCCSR)

Read/Write
Reset Value: (00h)

7	6	5	4	3	2	1	0
COCO	0	ADON	0	0	0	CH1	CH0

Bit 7 = COCO *Conversion Complete*

This bit is set by hardware. It is cleared by software by reading the result in the DR register or writing to the CSR register.

0 Conversion is not complete (default)

1 Conversion can be read from the DR register.

Bit 6 = Reserved. This bit must be cleared by software.

Bit 5 = ADON *A/D converter On*

This bit is set and cleared by software.

0 A/D converter is switched off (default)

1 A/D converter is switched on

Note: Remember that the ADC needs time to stabilize after the ADON bit is set.

Bits [4:2] = Reserved. Forced to 0 by hardware.

Bits [1:0] = CH[1:0] Channel Selection.

These bits are set and cleared by software. They select the analog input to be converted.

Table 20: Channel Selection

Pin	CH1	CH0
AIN0 (Default)	0	0
AIN1	0	1
AIN2	1	0
AIN3	1	1

DATA REGISTER (ADCDR)

Read Only
Reset Value: (00h)

7	6	5	4	3	2	1	0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Bits [7:0] = AD[7:0] Analog Converted Value.

This register contains the converted analog value in the range 00h to FFh.

Reading this register resets the COCO flag.

10 I²C Single-Master Bus Interface

10.1 Introduction

The I²C Bus Interface serves as an interface between the microcontroller and the serial I²C bus. It provides single-master functions, and controls all I²C bus-specific sequencing, protocol and timing. It supports Fast I²C mode (400 kHz) and up to 800 kHz for certain applications.

10.2 Main Features

- Parallel / I²C bus protocol converter
- Interrupt generation
- Standard I²C mode/Fast I²C mode (up to 800 kHz for certain applications)
- 7-bit Addressing

I²C Single Master Mode

- End of byte transmission flag
- Transmitter /Receiver flag
- Clock generation

10.3 General Description

In addition to receiving and transmitting data, this interface converts data from serial to parallel format and vice versa, using either an interrupt or a polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

Mode Selection

The interface can operate in the two following modes:

1. Master transmitter/receiver,
2. Idle (default).

The interface automatically switches from Idle to Master mode after it generates a START condition and from Master to Idle mode after it generates a STOP condition.

Communication Flow

The interface 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. Both start and stop conditions are generated by software.

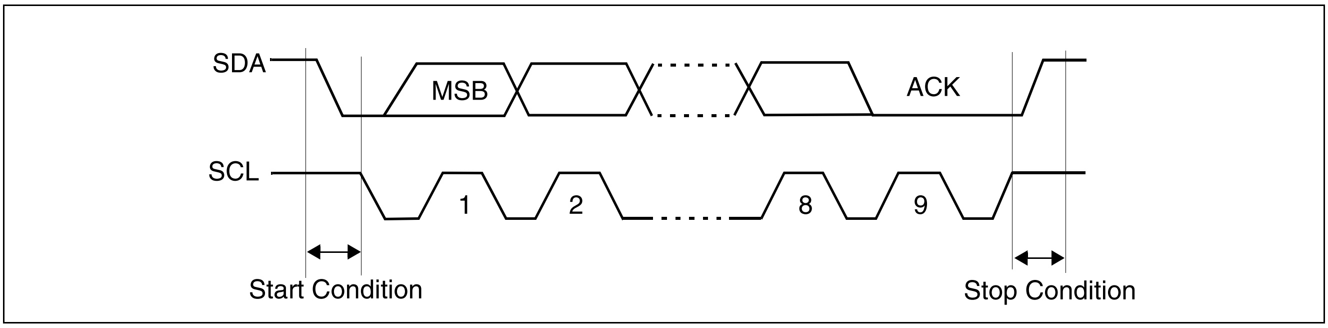
Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition is the address byte.

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 28.

Acknowledge is enabled and disabled by software.

The speed of the I²C interface is selected as Standard (0 to 100 kHz) and Fast I²C (100 to 400 kHz) and up to 800 kHz for certain applications.

Figure 28: I²C Bus Protocol



SDA/SCL Line Control

Transmitter mode: The interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the Data Register.

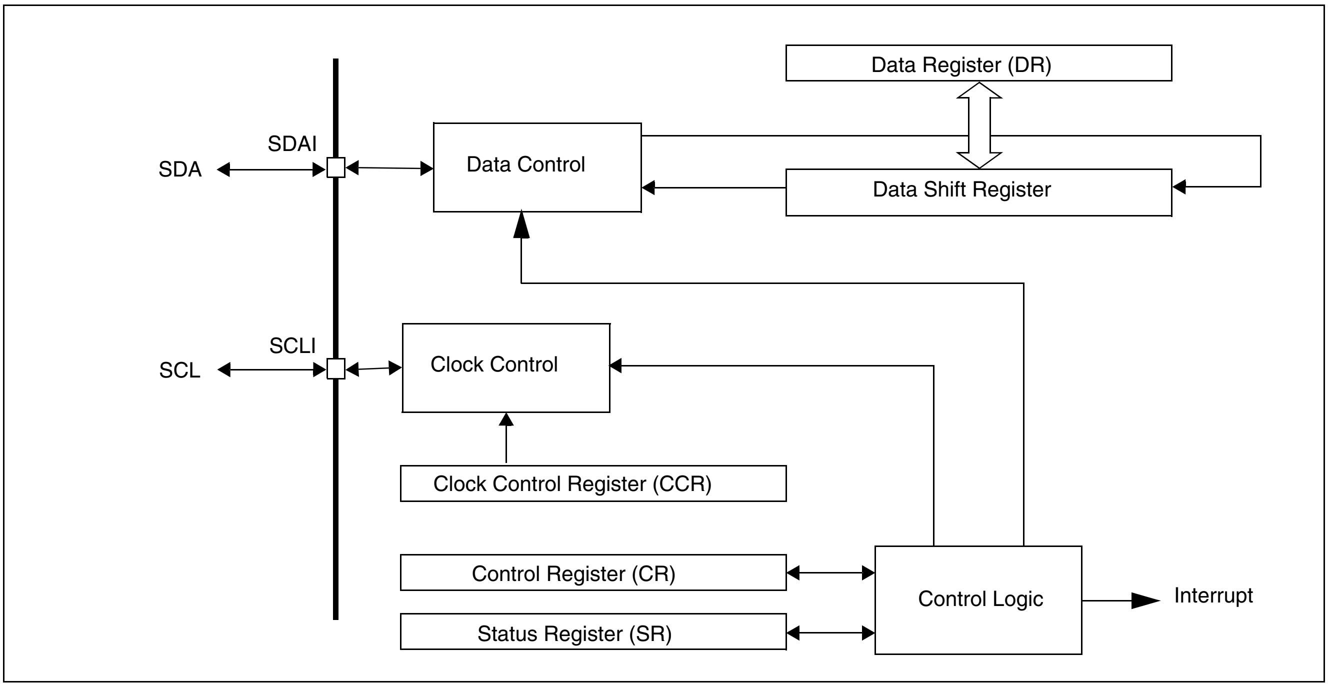
Receiver mode: The interface holds the clock line low after reception to wait for the microcontroller to read the byte in the Data Register.

The SCL frequency (f_{SCL}) is controlled by a programmable clock divider which depends on the I²C bus mode.

When the I²C cell is enabled, the SDA and SCL ports must be configured as a floating open-drain output or a floating input. In this case, the value of the external pull-up resistor used depends on the application.

When the I²C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 29: I²C Interface Block Diagram



10.4 Functional Description (Master Mode)

By default, the I²C interface operates in Idle mode (M/IDL bit is cleared) except when it initiates a transmit or receive sequence.

To switch from default Idle mode to Master mode a Start condition must be generated.

Setting the START bit causes the interface to switch to Master mode (M/IDL bit set) and generates a Start condition.

Once the Start condition is sent, the EVF and SB bits are set by hardware and an interrupt is generated if the ITE bit is set.

Then the master waits for a read of the SR register followed by a write in the DR register with the Slave address byte, holding the SCL line low (EV1).

Then the slave address byte is sent to the SDA line via the internal shift register.

After completion of this transfer (and the reception of an acknowledge from the slave if the ACK bit is set), the EVF bit is set by hardware and an interrupt is generated if the ITE bit is set.

Then the master waits for a read of the SR register followed by a write in the CR register (for example set PE bit), holding the SCL line low (EV2).

Next the master must enter Receiver or Transmitter mode.

10.5 Transfer Sequencing

10.5.1 Master Receiver

Following the address transmission and after SR and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- an Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR register followed by a read of the DR register, holding the SCL line low (EV3).

To close the communication, before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface automatically returns to Idle mode (M/IDL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

10.5.2 Master Transmitter

Following the address transmission and after SR register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR register followed by a write in the DR register, holding the SCL line low (EV4).

When the acknowledge bit is received, the interface sets the EVF and BTF bits with an interrupt if the ITE bit is set.

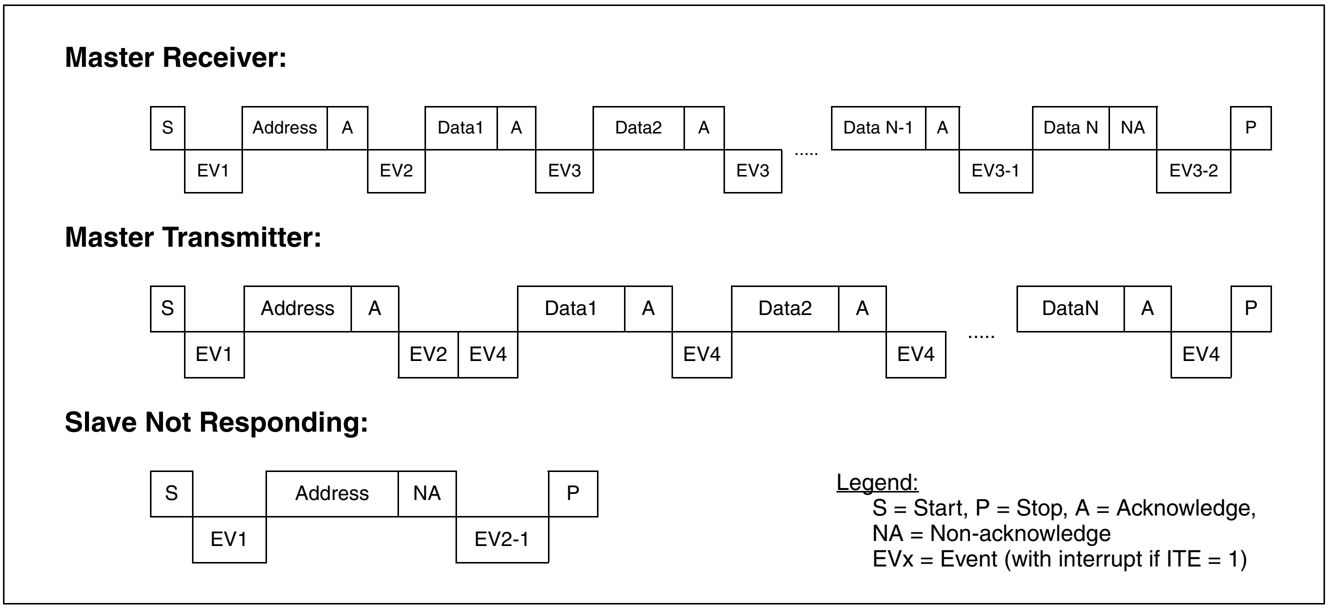
To close the communication, after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface automatically returns to Idle mode (M/IDL bit cleared).

Error Case:

AF: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the START or STOP bit.

Note: The SCL line is not held low if AF = 1.

Figure 30: Transfer Sequencing



EV1: EVF = 1, SB = 1, cleared by reading the SR register followed by writing to the DR register.

EV2: EVF = 1, cleared by reading the SR register followed by writing to the CR register (for example PE = 1).

EV2-1: EVF = 1, AF = 1, cleared by reading the SR register followed by writing STOP = 1 in the CR register.

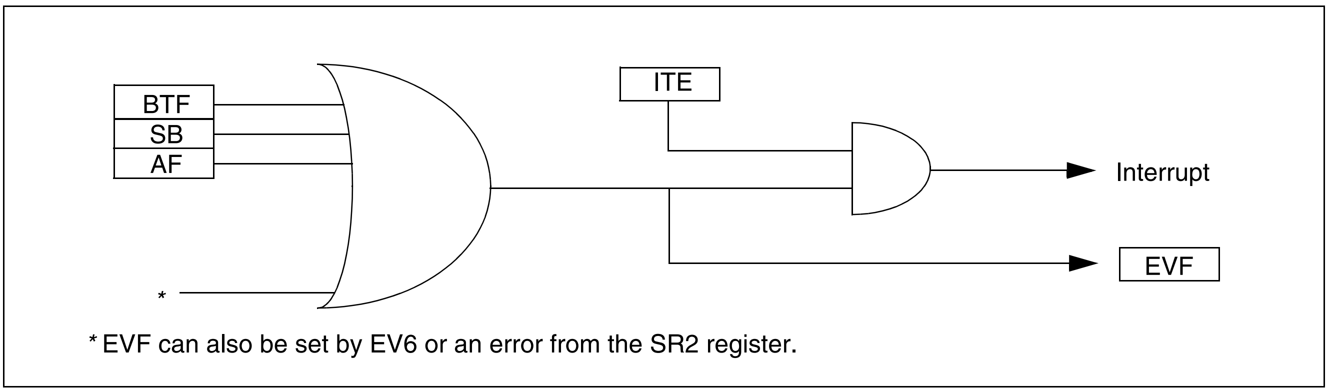
EV3: EVF = 1, BTF = 1, cleared by reading the SR register followed by reading the DR register.

EV3-1: Same as EV3, but ACK bit in CR register must be cleared before reading the DR register in order to send a NAK pulse after the “Data N” byte.

EV3-2: Same as EV3, but STOP = 1 must be written in the CR register.

EV4: EVF = 1, BTF = 1, cleared by reading the SR register followed by writing to the DR register.

Figure 31: Event Flags and Interrupt Generation



10.6 Register Description

Table 21: I²C Register Map

Addr. (Hex.)	Reset	R/W	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
001Ch	00h	R/W	I2CCR	00		PE	0	START	ACK	STOP	ITE
001Dh	00h	Read only	I2CSR	EVF	AF	TRA	0	BTF	0	M/IDL	SB
001Eh	00h	R/W	I2CCCR	FM/SM	FILTOFF	CC[5:0]					
001Fh	00h	R/W	I2CDR	DR7[:0]							

I²C CONTROL REGISTER (I2CCR)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	PE	0	START	ACK	STOP	ITE

Bits [7:6] = Reserved. Forced to 0 by hardware.

Bit 5 = PE *Peripheral enable.*
This bit is set and cleared by software.

0 Peripheral disabled

1 Master capability

Note: When PE = 0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released when PE = 0.
When PE = 1, the corresponding I/O pins are selected by hardware as alternate functions.
To enable the I²C interface, write the CR register ***TWICE*** *with PE = 1 as the first write only activates the interface (only PE is set).*

Bit 4 = Reserved. Forced to 0 by hardware

Bit 3 = START *Generation of a Start condition*. This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0) or when the Start condition is sent (with interrupt generation if ITE = 1).

In Master mode:

0 No start generation

1 Repeated start generation

In Idle mode:

0 No start generation

1 Start generation when the bus is free

Bit 2 = ACK *Acknowledge enable.*
This bit is set and cleared by software. Cleared by hardware when the interface is disabled (PE = 0).

0 No acknowledge returned

1 Acknowledge returned after an address byte or a data byte is received

Bit 1 = STOP *Generation of a Stop condition.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0) or when the Stop condition is sent. In Master mode only:

0 No stop generation

1 Stop generation after the current byte transfer or after the current Start condition is sent.

Bit 0 = ITE *Interrupt enable.*
This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE = 0).

0 Interrupt disabled

1 Interrupt enabled

I²C STATUS REGISTER (I2CSR)

Read Only
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EVF	AF	TRA	0	BTF	0	M/IDL	SB

Bit 7 = EVF *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software by reading the SR register in case of error event or as described in Section 10.5: Transfer Sequencing. It is also cleared by hardware when the interface is disabled (PE = 0).

0 No event

1 One of the following events has occurred:

BTF = 1 (Byte received or transmitted)

SB = 1 (Start condition generated)

AF = 1 (No acknowledge received after byte transmission if ACK = 1)

Address byte successfully transmitted.

Bit 6 = AF Acknowledge Failure.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE = 1. It is cleared by software by reading the SR register or by hardware when the interface is disabled (PE = 0).

The SCL line is not held low when AF = 1.

0 No acknowledge failure

1 Acknowledge failure

Bit 5 = TRA *Transmitter/Receiver.*
When BTF is set, TRA = 1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware when the interface is disabled (PE = 0).

0 Data byte received (if BTF = 1)

1 Data byte transmitted

Bit 4 = Reserved. *Forced to 0 by hardware.*

Bit 3 = BTF *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt

generation if ITE = 1. It is cleared by software by reading the SR register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE = 0).

Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV2 event (See Section 10.5: Transfer Sequencing). BTF is cleared by reading SR register followed by writing the next byte in DR register.

Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK = 1. BTF is cleared by reading SR register followed by reading the byte from DR register.

The SCL line is held low when BTF = 1.

0 Byte transfer not done

1 Byte transfer succeeded

Bit 2 = Reserved. *Forced to 0 by hardware*.

Bit 1 = M/IDL *Master/Idle.*
This bit is set by hardware when the interface is in Master mode (writing START = 1). It is cleared by hardware after a Stop condition on the bus. It is also cleared by hardware when the interface is disabled (PE = 0).

0 Idle mode

1 Master mode

Bit 0 = SB *Start bit.*
This bit is set by hardware when a Start condition is generated (following a write START = 1). An interrupt is generated if ITE = 1. It is cleared by software by reading the SR register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE = 0).

0 No Start condition

1 Start condition generated

I²C CLOCK CONTROL REGISTER (I2CCCR)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
FM/SM	FILTOFF	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = FM/SM *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE = 0).

0 Fast I²C mode

1 Standard I²C mode

Bit 6 = FILTOFF *Filter Off.*
This bit is set and cleared by software, it is not taken into account in the EMU version and is considered as always set to 1 (inactive filter).

When set, it disables the filter of the I²C pads in order to achieve speeds of over 400 kHz on a short-length I²C bus (at the user's responsibility). Such high frequencies are computed with the Fast mode formula given below.

Bits [5:0] = CC[5:0] *6-bit clock divider.*
These bits select the speed of the bus (f_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE = 0). The value of the 6-bit clock divider, CC[5:0] ≥ 03h

Fast mode (FM/SM = 0): f_{SCL} > 100 kHz

$f_{SCL} = f_{CPU}/([2x([CC5...CC0]+3)]+1)$

Standard mode (FM/SM = 1): f_{SCL} ≤100 kHz

$f_{SCL} = f_{CPU}/(3x([CC5...CC0]+3))$

Note: The programmed f_{SCL} speed assumes that there is no load on the SCL and SDA lines.

I²C DATA REGISTER (I2CDR)

Read / Write
Reset Value: 0000 0000 (00h)

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

Bits [7:0] = D[7:0] *8-bit Data Register*.
These bits contain the byte to be received or transmitted on the bus.

Transmitter mode: Bytes are automatically transmitted when the software writes to the DR register.

Receiver mode: The first data byte is automatically received in the DR register using the least significant bit of the address.

Then, the subsequent data bytes are received one-by-one after reading the DR register.

11 Display Data Channel Interfaces (DDC)

11.1 Introduction

The DDC (Display Data Channel) bus interfaces are mainly used by the monitor to identify itself to the video controller, by the monitor manufacturer to perform factory alignment, and by the user to adjust the monitor's parameters. Both DDC interfaces consist of:

- A fully hardware-implemented interface, supporting DDC2B (VESA specification 3.0 compliant). It accesses the ST7 on-chip memory directly through a built-in DMA engine.
- A second interface, supporting the slave I²C functions for handling DDC/CI mode (DDC2Bi), factory alignment, HDCP, Enhanced DDC (EDDC) or other addresses by software.

Each DDC interface has its own dedicated DMA area in RAM. In the event of concurrent DMA accesses, the DDC A cell has priority over the DDC B cell.

11.2 DDC Interface Features

11.2.1 Hardware DDC2B Interface Features

- Full hardware support for DDC2B communications (VESA specification version 3)
- Hardware detection of DDC2B addresses A0h/A1h
- Separate mapping of EDID version 1: Base (128 bytes) and Extended (128 bytes)
- Support for error recovery mechanism
- Detection of misplaced Start and Stop conditions
- Random and Sequential I²C byte read modes
- DMA transfer from any memory location and to RAM
- Automatic memory address increment
- End of data downloading flag, end of communication flag and interrupt capability

11.2.2 DDC/CI Factory Interface Features

General I²C Features

- Parallel bus /I²C protocol converter
- Interrupt generation
- Standard I²C mode
- 7-bit Addressing

I²C Slave Features

- I²C bus busy flag
- Start bit detection flag
- Detection of misplaced Start or Stop condition
- Transfer problem detection
- Address Matched detection
- 2 Programmable Address detection and/or Hardware detection of DDC/CI addresses (6Eh/6Fh)
- End of byte transmission flag
- Transmitter/Receiver flag
- Stop condition Detection

Figure 32: DDC Interface Overview

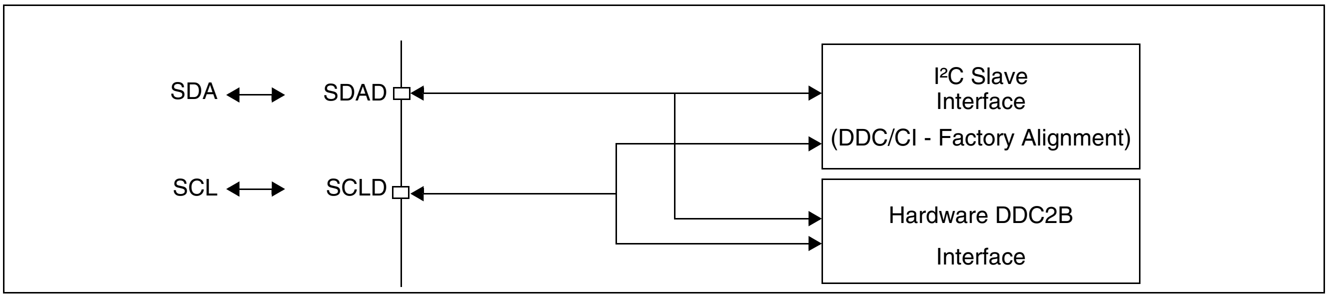
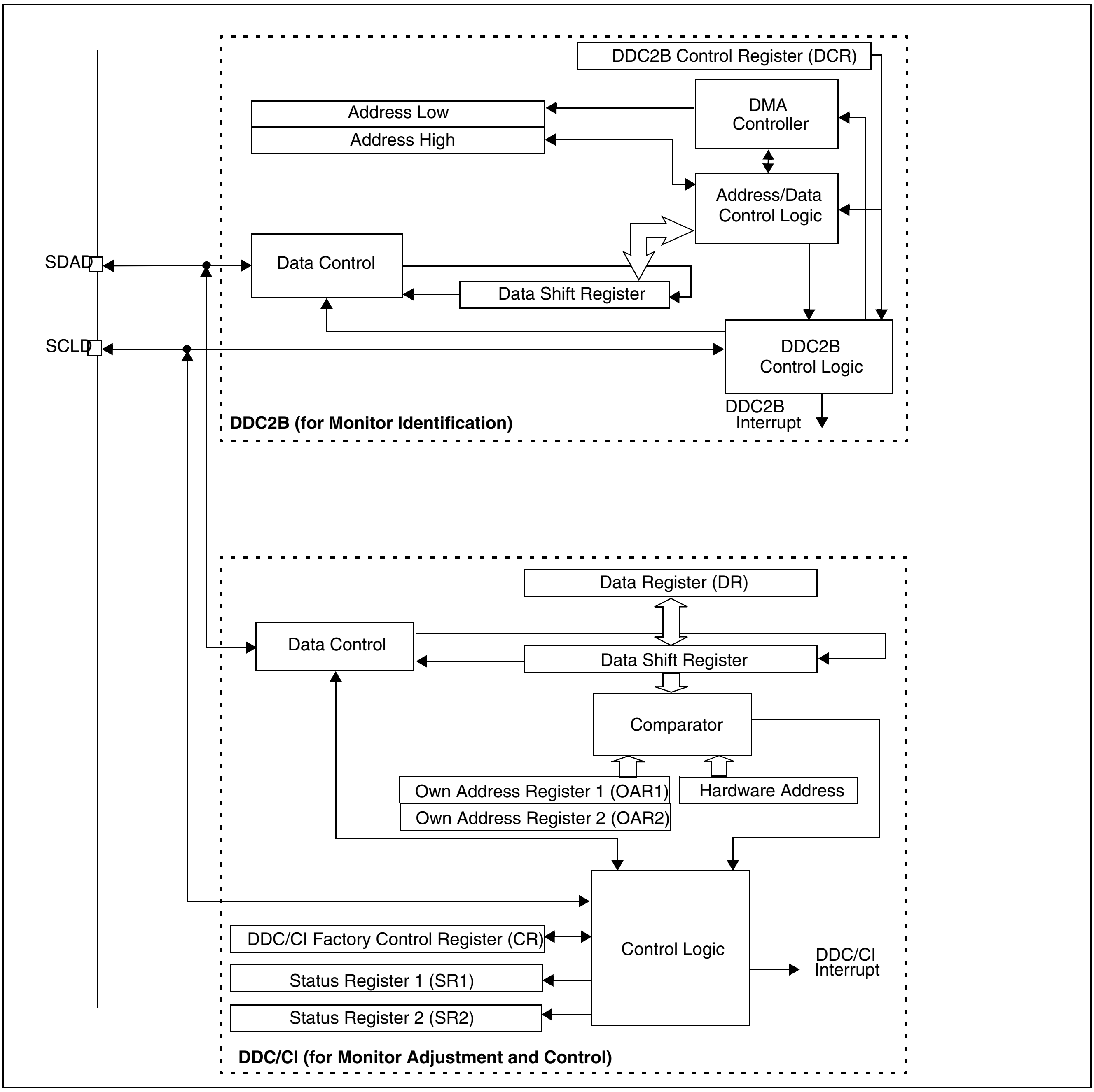


Figure 33: DDC Interface Block Diagram



11.3 Signal Description

11.3.1 Serial Data (SDA)

The SDA bidirectional pin is used to transfer data in and out of the device. An external pull-up resistor must be connected to the SDA line. Its value depends on the load of the line and the transfer rate.

11.3.2 Serial Clock (SCL)

The SCL input pin is used to synchronize all data in and out of the device when in I²C bidirectional mode. An external pull-up resistor must be connected to the SCL line. Its value depends on the load of the line and the transfer rate.

Note: When the DDC2B and DDC/CI Factory Interfaces are disabled (HWPE bit = 0 in the DCR register and PE bit = 0 in the CR register), the SDA and SCL pins revert to being standard I/O pins.

11.4 DDC Standard

The DDC standard is divided into several data transfer protocols: DDC2B, DDC/CI and other slave communication standards (HDCP, E-DDC, etc.).

For DDC2B, refer to the “VESA DDC Standard v3.0” specification. For DDC/CI refer to the “VESA DDC Commands Interface v1.0”

DDC2B is a unidirectional channel from display to host. The host computer uses base-level I²C commands to read the EDID data from the display which is always in Slave mode.

DDC/CI is a bidirectional channel between the host computer and the display. The DDC/CI offers a display control interface based on I²C bus. Only the DDC2Bi interface is supported (and not the DDC2B+ or DDC2AB interfaces).

11.4.1 DDC2B Interface

The DDC2B Interface acts as an I/O interface between a DDC bus and the MCU memory. In addition to receiving and transmitting serial data, this interface directly transfers parallel data to and from memory using a DMA engine, only halting CPU activity for 2 clock cycles during each byte transfer.

The interface supports the following by hardware:

- DDC2B communication protocol
- write operations into RAM
- read operations from RAM

In DDC2B mode, it operates in I²C Slave mode.

Device addresses A0h/A1h are recognized. EDID version 1 is used.

The Write and Read operations allow the EDID data to be downloaded during factory alignment (for example).

Writing to the memory by the DMA engine is inhibited by the WP bit in the DCR register. A write of the last data structure byte sets a flag and may be programmed to generate an interrupt request.

The Data address (sub-address) is either the **second** byte of write transfers or is pointed to by the internal address counter which automatically increments after each byte transfer. The physical address mapping of the data structure is fixed by hardware in a dedicated RAM area (see Table 24:

EDID DMA Pointer Configuration).

11.4.2 Mode Description

DDC2B Mode: The DDC2B Interface enters DDC2B mode from the initial state if the software sets the HWPE bit. Once in DDC2B mode, the Interface always acts as a slave following the protocol described in Figure 34.

The DDC2B Interface continuously monitors the SDA and SCL lines for a START condition and will not respond (no acknowledge) until one is found.

A STOP condition at the end of a Read command (after a NACK) forces the stand-by state. A STOP condition at the end of a Write command triggers the internal DMA write cycle.

The Interface samples the SDA line on the rising edge of the SCL signal and outputs data on the falling edge of the SCL signal. In any case, the SDA line can only change when the SCL line is low.

Figure 34: DDC2B Protocol Example

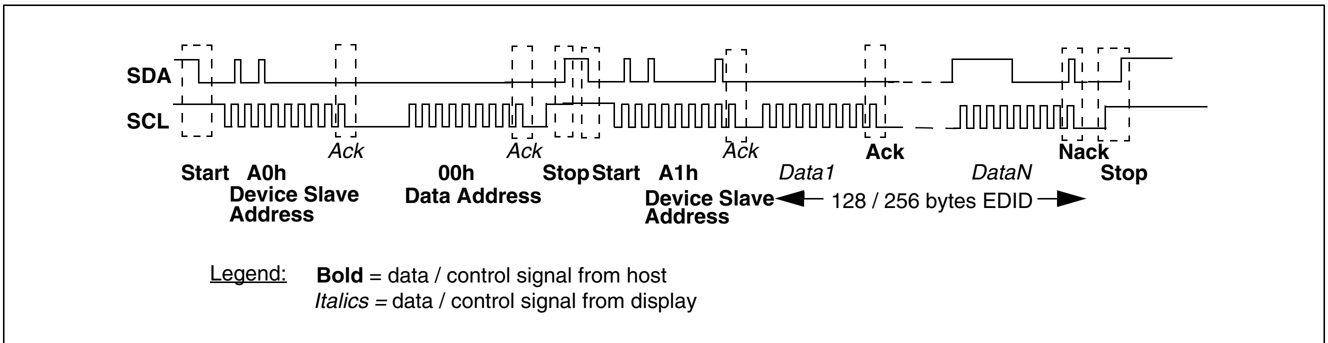
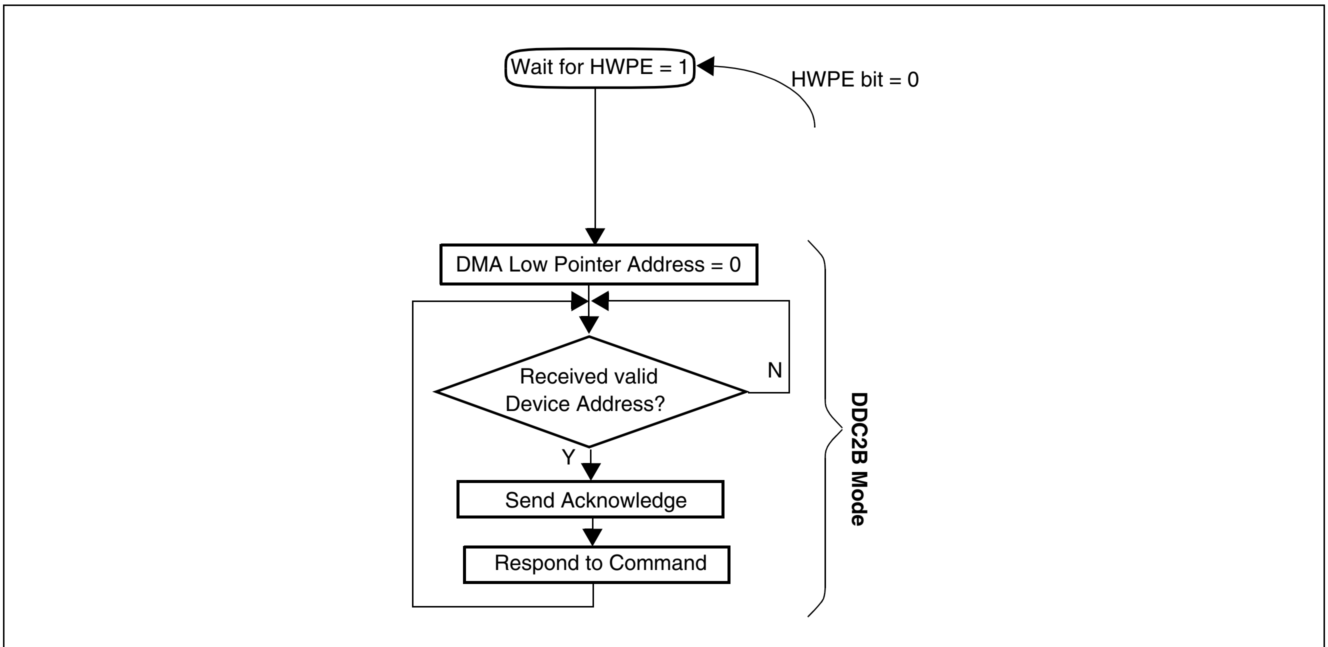


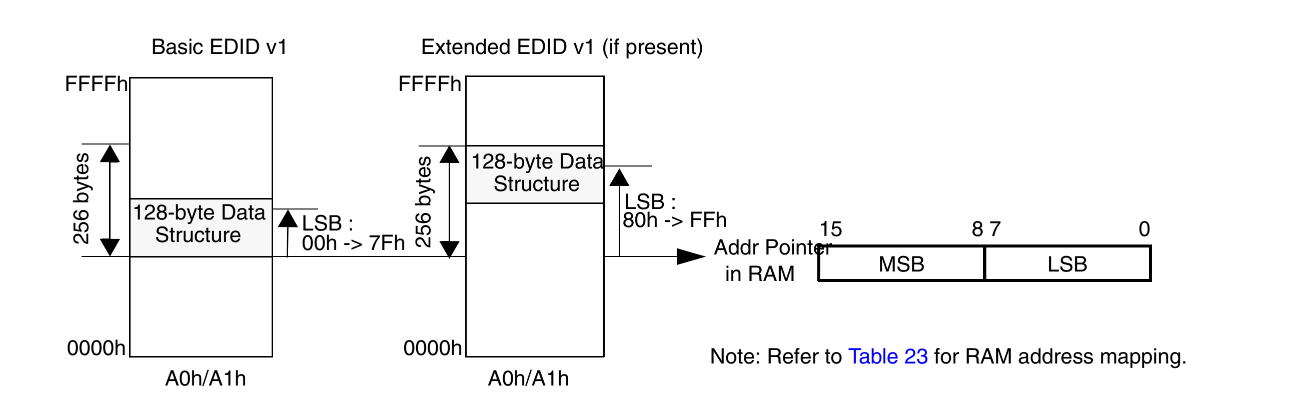
Figure 35: DDC1/2B Operation Flowchart



EDID Data structure mapping: An internal address pointer defines the memory location being addressed.

It defines the 256-byte block within the RAM address space containing the data structure. The LSB is loaded with the data address sent by the master after a write Device Address. It defines the byte within the data structure currently addressed. It is reset upon entry into the DDC2B mode.

Figure 36: Mapping of DDC2B Data Structure



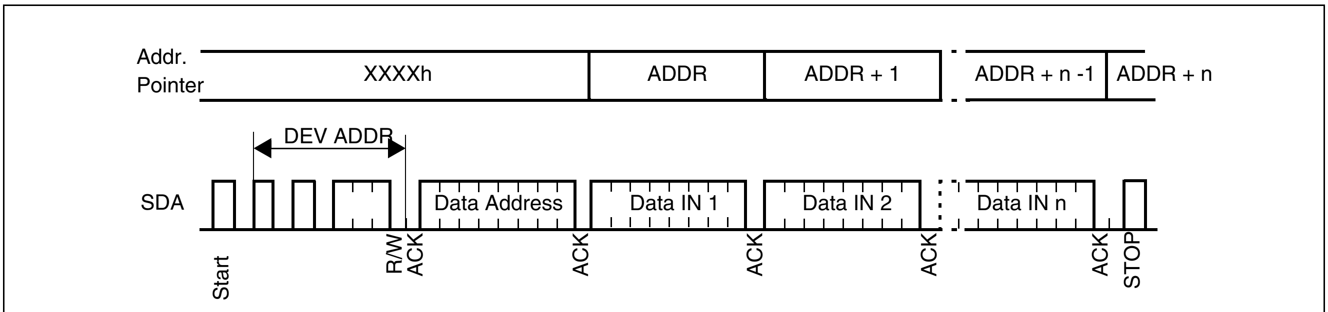
Write Operation

Once the DDC2B Interface has acknowledged a write transfer request, i.e. a Device Address with RW = 0, it waits for a data address. When the latter is received, it is acknowledged and loaded into the LSB.

Then, the master may send any number of data bytes that are all acknowledged by the DDC2B Interface. The data bytes are written in RAM if the WP bit = 0 in the DCR register, otherwise the RAM location is not modified.

Write operations are always performed in RAM and therefore do not delay DDC transfers. Meanwhile, concurrent software execution is halted for 2 clock cycles.

Figure 37: Write Sequence



Read Operations

All read operations consist of retrieving the data pointed to by an internal address counter which is initialized by a dummy write and which increments with any read. The DDC2B Interface always waits for an acknowledge during the 9th bit-time. If the master does not pull the SDA line low during this bit-time, the DDC2B Interface ends the transfer and switches to a stand-by state.

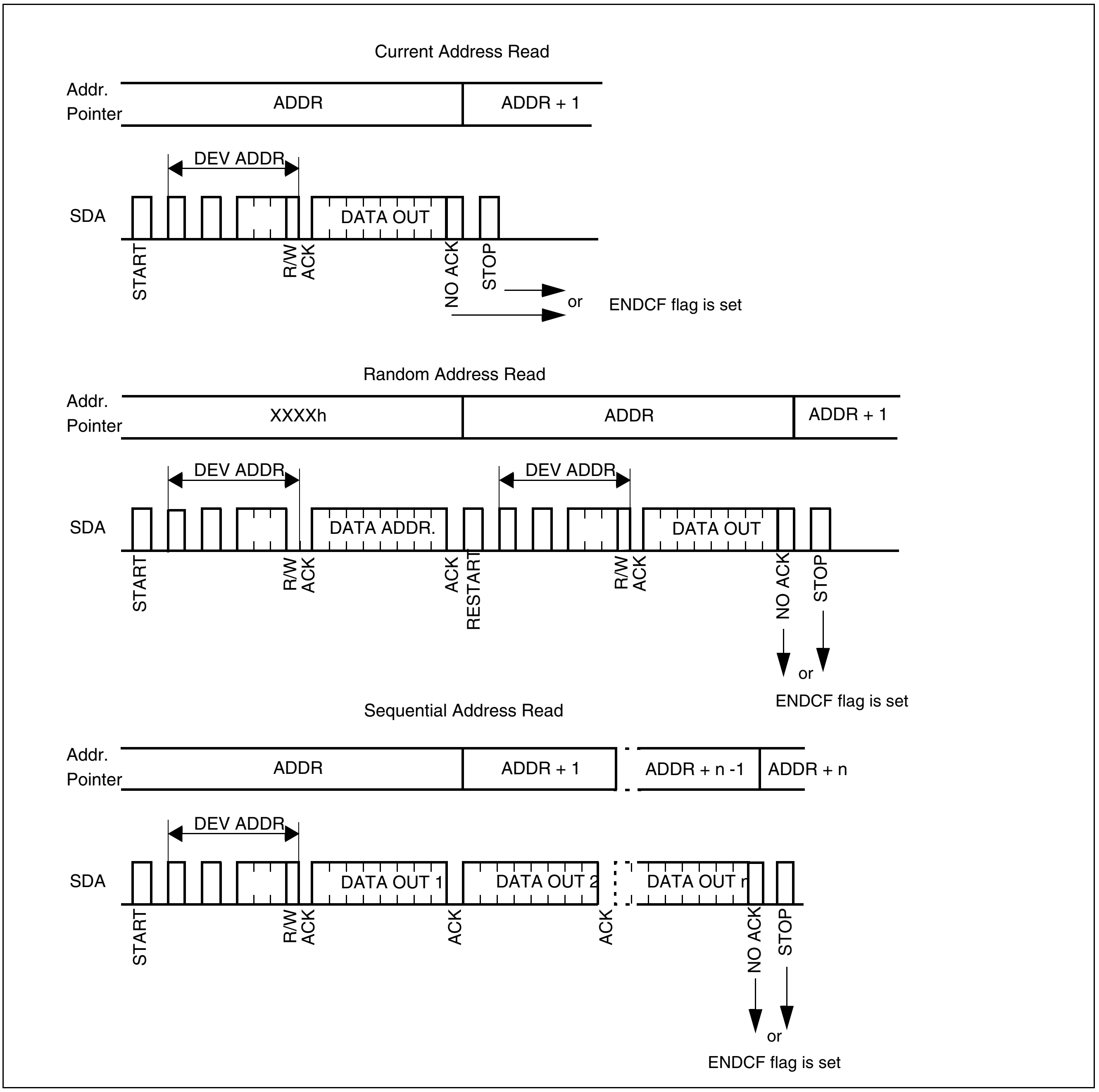
Current address read: After generating a START condition the master sends a read device address (RW = 1). The DDC2B Interface acknowledges this and outputs the data byte pointed to by the internal address pointer which subsequently increments. The master must NOT acknowledge this byte and must terminate the transfer with a STOP condition.

Random address read: The master performs a dummy write to load the data address into the pointer LSB. Then the master sends a RESTART condition followed by a read Device Address (RW = 1).

Sequential address read: This mode is similar to the current and random address reads, except that the master DOES acknowledge the data byte for the DDC2B Interface to output the next byte in sequence. To terminate the read operation the master must NOT acknowledge the last data byte and must generate a STOP condition. The data output are issued from consecutive memory addresses.

End of communication: Upon a detection of NACK or STOP conditions at the end of a read transfer, the bit ENDCF is set and an interrupt is generated if ENDCE is set.

Figure 38: Read Sequences



Read and Write Operations

After each byte transfer, the internal address counter automatically increments. If the counter is pointing to the top of the structure, it rolls over to the bottom since the increment is performed only on the 7 or 8 LSBs of the pointer depending on the selected data structure size. It rolls over from 7Fh to 00h or from FFh to 80h depending on the MSB of the last data address received.

Then after that last byte has been effectively written or read in RAM at LSB address 7Fh or FFh, the EDF flag is set and an interrupt is generated if EDE is set.

The transfer is terminated by the master generating a STOP condition.

11.5 DDC/CI Factory Alignment Interface

Refer to the CR, SR1 and SR2 registers in Section 11.7: Register Description for the bit definitions.

The DDC/CI interface works as an I/O interface between the microcontroller and the DDC2Bi, HDCP, E-DDC or Factory alignment protocols. It receives and transmits data in Slave I²C mode using an interrupt or polled handshaking.

The interface is connected to the I²C bus through a data pin (SDAD) and a clock pin (SCLD) configured as an open-drain output.

The DDC/CI interface has five internal register locations. Two of them are used to initialize the interface:

1. 2 Own Address Registers OAR1 and OAR2
2. Control register CR

The following four registers are used during data transmission/reception:

1. Data Register DR
2. Control Register CR
3. Status Register 1 SR1
4. Status Register 2 SR2

The interface decodes an I²C or DDC2Bi address stored by software in either OAR register and/or the DDC/CI address (6Eh/6Fh) as its default hardware address.

After a reset, the interface is disabled.

11.5.1 I²C Modes

The interface operates in Slave Transmitter/Receiver modes.

The master generates both Start and Stop conditions. The I²C clock (SCL) is always received by the interface from a master, but the interface is able to stretch the clock line.

The interface can recognize its two programmable addresses (7-bit) and its default hardware address (DDC/CI address: 6Eh/6Fh). The DDC/CI address detection may be enabled or disabled by software. It never recognizes the Start byte (01h) whatever its own address is.

Slave mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register where it is compared to the programmable addresses or to the DDC/CI address (if selected by software).

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the following events occur in sequence:

- Acknowledge pulse is generated if the ACK bit is set.
- EVF and ADSL bits are set.
- An interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see EV1 in Section 11.6: Transfer Sequencing). Next, the DR register must be read to determine from the least significant bit if the slave must enter Receiver or Transmitter mode.

Slave Receiver

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte, the following events occur in sequence:

- an Acknowledge pulse is generated if the ACK bit is set.
- the EVF and BTF bits are set.
- an interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see EV2 in Section 11.6: Transfer Sequencing).

Slave Transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see EV3 in Section 11.6: Transfer Sequencing).

When the acknowledge pulse is received:

- the EVF and BTF bits are set.
- an interrupt is generated if the ITE bit is set.

Closing Slave Communication

After the last data byte is transferred, a Stop Condition is generated by the master. The interface detects this condition and in this case:

- the EVF and STOPF bits are set.
- an interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see EV4 in Section 11.6: Transfer Sequencing).

Error Cases

BERR: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set and an interrupt is generated if the ITE bit is set.

If it is a Stop condition, then the interface discards the data, releases the lines and waits for another Start condition. If it is a Start condition, then the interface discards the data and waits for the next slave address on the bus.

AF: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set and an interrupt is generated if the ITE bit is set.

Note: In both cases, the SCL line is not held low. However, the SDA line can remain low due to possible ‘0’ bits transmitted last. It is then necessary to release both lines by software.

How to Release the SDA / SCL Lines

Set and subsequently clear the STOP bit when BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

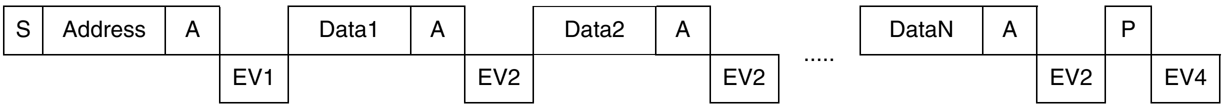
Other Events

ADSL: Detection of a Start condition after an acknowledge time-slot. The state machine is reset and starts a new process. The ADSL bit is set and an interrupt is generated if the ITE bit is set. The SCL line is stretched low.

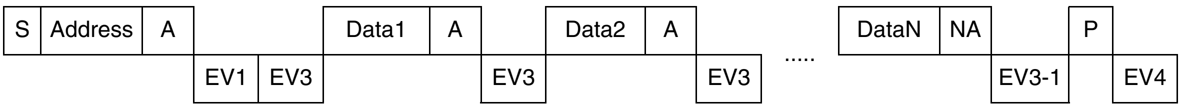
STOPF: Detection of a Stop condition after an acknowledge time-slot. The state machine is reset. Then the STOPF flag is set and an interrupt is generated if the ITE bit is set.

11.6 Transfer Sequencing

Slave Receiver



Slave Transmitter



Legend:
S = Start, P = Stop, A = Acknowledge, NA = Non-acknowledge and EVx = Event (with interrupt if ITE = 1)

EV1: EVF = 1, ADSL = 1, cleared by reading register SR1.

EV2: EVF = 1, BTF = 1, cleared by reading register SR1 followed by reading DR register.

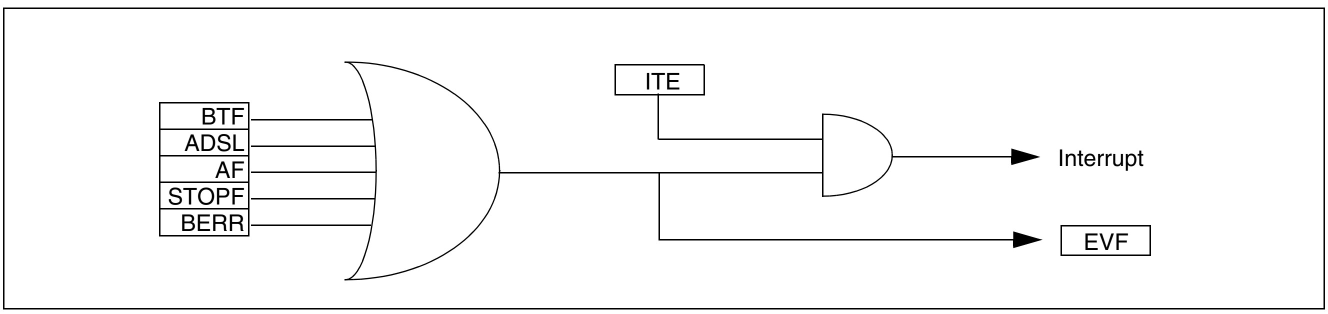
EV3: EVF = 1, BTF = 1, cleared by reading register SR1 followed by writing DR register.

EV3-1: EVF = 1, AF = 1 and BTF = 1, AF is cleared by reading register SR2, BTF is cleared by releasing the lines (write STOP = 1, STOP = 0 in register CR) or by writing to register DR (DR = FFh).

Note: If the lines are released by STOP = 1, STOP = 0, the subsequent EV4 is not seen.

EV4: EVF = 1, STOPF = 1, cleared by reading register SR2.

Figure 39: Event Flags and Interrupt Generation



11.7 Register Description

Table 22: DDCA Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0020h	00h	R/W	DDCCRA	0	0	PE	DDCCIEN	0	ACK	STOP	ITE
0021h	00h	R	DDCSR1A	EVF	0	TRA	BUSY	BTF	ADSL	0	0
0022h	00h	R	DDCSR2A	0	0	0	AF	STOPF	0	BERR	DDCIF
0023h	00h	R/W	DDCOAR1A	ADD[7:1]							0
0024h	00h	R/W	DDCOAR2A	ADD[7:1]							0
0025h	00h	R/W	DDCDRA	DR[7:0]							
0026h	00h	R/W	Reserved								
0027h	00h	R/W	DDCDCRA	0	0	ENDCF	ENDCE	EDF	EDE	WP	DDC2BPE

Table 23: DDCB Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0028h	00h	R/W	DDCCRB	0	0	PE	DDCCIEN	0	ACK	STOP	ITE
0029h	00h	R	DDCSR1B	EVF	0	TRA	BUSY	BTF	ADSL	0	0
002Ah	00h	R	DDCSR2B	0	0	0	AF	STOPF	0	BERR	DDCIF
002Bh	00h	R/W	DDCOAR1B	ADD[7:1]							0
002Ch	00h	R/W	DDCOAR2B	ADD[7:1]							0
002Dh	00h	R/W	DDCDRB	DR[7:0]							
002Eh	00h	R/W	Reserved								
002Fh	00h	R/W	DDCDCRB	0	0	ENDCF	ENDCE	EDF	EDE	WP	DDC2BPE

Table 24: EDID DMA Pointer Configuration

Cell	Basic EDID	Extended EDID
DDCA	600h to 67Fh	680h to 6FFh
DDCB	700h to 77Fh	780h to 7FFh

DDC CONTROL REGISTER (DDCCR)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	PE	DDCCIEN	0	ACK	STOP	ITE

Bits [7:6] = Reserved. Forced to 0 by hardware.

Bit 5 = PE *DDC/CI Peripheral enable.*
This bit is set and cleared by software.

0 Peripheral disabled

1 Peripheral enabled

Note: When PE = 0, all the bits of the CR, SR1 and SR2 registers are reset. All outputs are released when PE = 0

When PE = 1, the corresponding I/O pins are selected by hardware as alternate functions.

To enable the I²C interface, write the CR register ***TWICE*** *with PE = 1 as the first write only activates the interface (only PE is set).*

Bit 4 = DDCCIEN *DDC/CI address detection enabled.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). The 6Eh/6Fh DDC/CI address is acknowledged.

0 DDC/CI address detection disabled

1 DDC/CI address detection enabled

Bit 3 = Reserved. Forced to 0 by hardware.

Bit 2 = ACK *Acknowledge enable.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0).

0 No acknowledge returned

1 Acknowledge returned after an address byte or a data byte is received

Bit 1 = STOP *Release I²C bus.*
This bit is set and cleared by software or when the interface is disabled (PE = 0).

Slave Mode:

0 Nothing

1 Release the SCL and SDA lines after the current byte transfer (BTF = 1). The STOP bit has to be cleared by software.

Bit 0 = ITE *Interrupt enable.*
This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE = 0).

0 Interrupt disabled

1 Interrupt enabled

Refer to Figure 39 for the relationship between the events and the interrupt.

SCL is held low when the BTF or ADSL is detected.

DDC STATUS REGISTER 1 (DDCSR1)

Read Only

Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EVF	0	TRA	BUSY	BTF	ADSL	0	0

Bit 7 = EVF *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software by reading the SR2 register in case of an error event or as described in Figure 39. It is also cleared by hardware when the interface is disabled (PE = 0).

0 No event

1 One of the following events has occurred:

BTF = 1 (Byte received or transmitted)

ADSL = 1 (Either address matched in Slave mode when ACK = 1)

AF = 1 (No acknowledge received after byte transmission if ACK = 1)

STOPF = 1 (Stop condition detected in Slave mode)

BERR = 1 (Bus error, misplaced Start or Stop condition detected)

Bit 6 = Reserved. Forced to 0 by hardware.

Bit 5 = TRA *Transmitter/Receiver.*
When BTF is set, TRA = 1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after a Stop condition (STOPF = 1) is detected or when the interface is disabled (PE = 0).

0 Data byte received (if BTF = 1)

1 Data byte transmitted

Bit 4 = BUSY *Bus busy.*
This bit is set by hardware on detection of a Start condition and cleared by hardware when a Stop condition is detected. It indicates that a communication is in progress on the bus. This information is still updated when the interface is disabled (PE = 0).

0 No communication on the bus

1 Communication ongoing on the bus

Bit 3 = BTF *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE = 1. It is cleared by software by reading the SR1 register followed by a read or a write to the DR register. It is also cleared by hardware when the interface is disabled (PE = 0).

Following a byte transmission, this bit is set after reception of the acknowledge clock pulse BTF is cleared by reading the SR1 register followed by writing the next byte in the DR register.

Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK = 1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low when BTF = 1.

0 Byte transfer not completed

1 Byte transfer succeeded

Bit 2 = ADSL *Address matched (Slave mode).* This bit is set by hardware as soon as the received slave address matched with the OARx registers content or the DDC/CI address is recognized. An interrupt is generated if ITE = 1. It is cleared by software by reading the SR1 register or by hardware when the interface is disabled (PE = 0).

The SCL line is held low when ADSL = 1.

0 Address mismatched or not received

1 Received address matched

Bits [1:0] = Reserved. Forced to 0 by hardware.

DDC STATUS REGISTER 2 (DDCSR2)

Read Only
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	0	AF	STOPF	0	BERR	DDCIF

Bits [7:5] = Reserved. Forced to 0 by hardware.

Bit 4 = AF *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE = 1. It is cleared by software by reading the SR2 register or by hardware when the interface is disabled (PE = 0).

The SCL line is not held low when AF = 1.

0 No acknowledge failure

1 Acknowledge failure

Bit 3 = STOPF *Stop detection.*
This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK = 1). An interrupt is generated if ITE = 1. It is cleared by software by reading the SR2 register or by hardware when the interface is disabled (PE = 0). The SCL line is not held low when STOPF = 1.

0 No Stop condition detected

1 Stop condition detected

Bit 2 = Reserved. Forced to 0 by hardware.

Bit 1 = BERR *Bus error.*
This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE = 1. It is cleared by software by reading the SR2 register or by hardware when the interface is disabled (PE = 0).

The SCL line is not held low when BERR = 1.

0 No misplaced Start or Stop condition

1 Misplaced Start or Stop condition

Bit 0 = DDCIF *DDC/CI address detected.*
This bit is set by hardware when the DDC/CI address (6Eh/6Fh) is detected on the bus when DDCIEN = 1. It is cleared by hardware when a Stop condition (STOPF = 1) is detected, or when the interface is disabled (PE = 0).

0 No DDC/CI address detected on bus

1 DDC/CI address detected on bus

DDC DATA REGISTER (DDCDR)

Read / Write
Reset Value: 0000 0000 (00h)

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

Bits [7:0] = D[7:0] *8-bit Data Register.*
These bits contain the byte to be received or transmitted on the bus.

Transmitter mode: Bytes are automatically transmitted when the software writes to the DR register.

Receiver mode: The first data byte is automatically received in the DR register using the least significant bit of the address. Then, the next data bytes are received one by one after reading the DR register.

DDC OWN ADDRESS REGISTER 1 (DDCOAR1)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	0

Bits [7:1] = ADD[7:1] *Interface address*.
These bits define the I²C bus programmable address of the interface. They are not cleared when the interface is disabled (PE = 0).

Bit 0 = Reserved. Forced to 0 by hardware.

DDC OWN ADDRESS REGISTER 2 (DDCOAR2)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	0

Bits [7:1] = ADD[7:1] *Interface address*.
These bits define the I²C bus programmable address of the interface. They are not cleared when the interface is disabled (PE = 0).

Bit 0 = Reserved. Forced to 0 by hardware.

DDC2B CONTROL REGISTER (DDCDCR)

Read / Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	ENDCF	ENDCE	EDF	EDE	WP	DDC2BPE

Bits [7:6] = Reserved. Forced by hardware to 0.

Bit 5 = ENDCF *End of Communication interrupt Flag*.
This bit is set by hardware. An interrupt is generated if ENDCE = 1. It must be cleared by software.

0 NACK or STOP condition not met in Read mode.

1 NACK or STOP condition met in Read mode.

Bit 4 = ENDCE *End of Communication interrupt Enable*.
This bit is set and cleared by software.

0 End of Communication interrupt disabled.

1 End of Communication interrupt enabled.

Bit 3 = EDF *End of Download interrupt Flag*.
This bit is set by hardware. An interrupt is generated if EDE = 1. It must be cleared by software.

0 Download not started or not completed yet.

1 Download completed. Last byte of data structure (relative address 7Fh or FFh) has been stored or read in RAM.

In Read Mode: EDF is set upon reading the next byte after the internal address counter has rolled over from 7Fh to 00h, or FFh to 80h.

In Write Mode: EDF is set when the last byte of data structure has been stored in RAM, and only if writing to the RAM is enabled (bit WP = 0). if writing occurs but WP=1, EDF is not set.

Bit 2 = EDE *End of Download interrupt Enable*.
This bit is set and cleared by software.

0 End of Download interrupt disabled.

1 End of Download interrupt enabled.

Bit 1 = WP *Write Protect*.
This bit is set and cleared by software.

0 Enable writes to the RAM.

1 Disable DMA write transfers and protect the RAM content.
CPU writes to the RAM are not affected.

Bit 0 = DDC2BPE *DDC2B Peripheral Enable*.
This bit is set and cleared by software.

0 Release the SDA port pin and ignore SCL port pin. The other bits of the DCR are left unchanged.

1 Enable the DDC Interface and respond to the DDC2B protocol.

Note: When DDC2BPE = 1, all the bits of the DCR register are locked and cannot be changed. The desired configuration therefore must be written in the DCR register with DDC2BPE = 0 and then set the DDC2BPE bit in a second step.

12 Watchdog Timer (WDG)

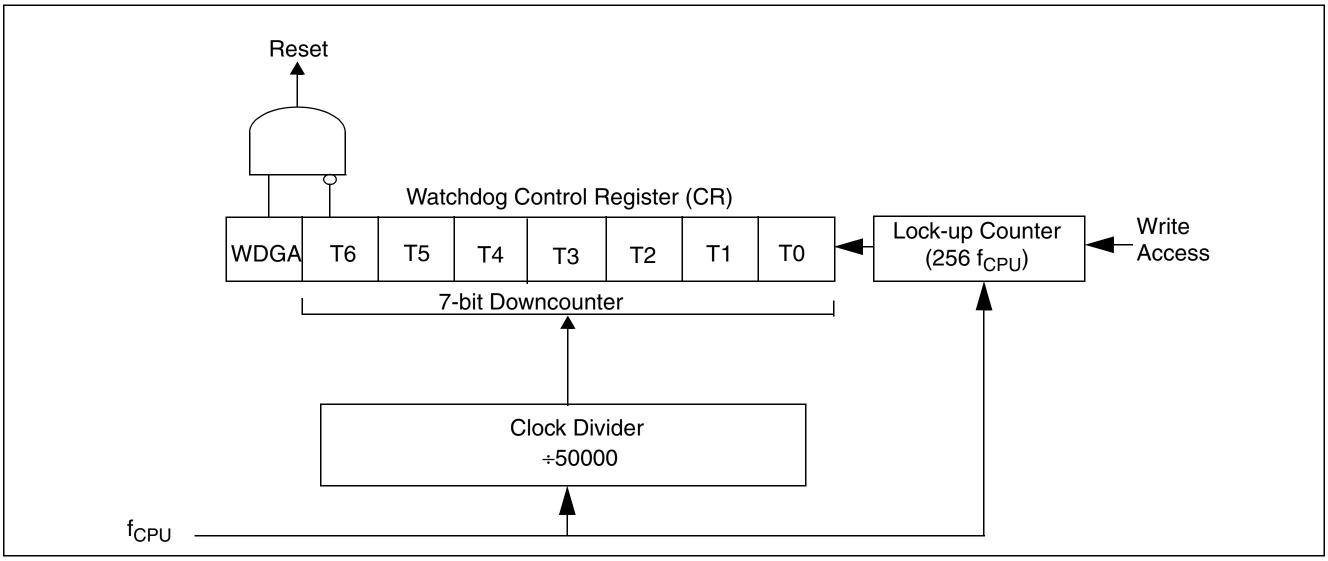
12.1 Introduction

The Watchdog Timer 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. The Watchdog circuit generates an MCU reset when the programmed time period expires, unless the program refreshes the counter's contents before the T6 bit is cleared. In addition, a second counter prevents the Watchdog register from being updated at intervals that are too close.

12.2 Main Features

- Programmable timer (64 increments of 50000 CPU cycles)
- Programmable reset
- Reset (if watchdog enabled) when the T6 bit reaches zero
- Reset (if watchdog enabled) on HALT instruction
- Lock-up Counter for preventing short time refreshes

Figure 40: Watchdog Block Diagram



12.3 Main Watchdog Counter

The counter value stored in the CR register (bits T[6:0]), is decremented every 50000 clock cycles, and the length of the time out period can be programmed by the user in 64 increments.

If the watchdog is enabled (bit WDGA is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 is cleared), it initiates a reset cycle pulling low the reset pin for typically 500 ns:

- The WDGA bit is set (watchdog enabled)
- Bit T6 is set to prevent generating an immediate reset
- Bits T[5:0] contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 25).

12.4 Lock-up Counter

An 8-bit counter starts after a reset or by writing to the CR register. It disables the writing of the CR register during the next 256 cycles of CPU clock (typical value of 32 µs at 8 MHz). If a writing order takes place during this time, this 8-bit counter is reset but not the main watchdog downcounter (no writing to the CR register occurs).

Thus after several too close writings of the CR register, the main downcounter reaches the reset value and a reset occurs. If the CR register is normally refreshed every 32 µs or more, write commands are always enabled.

Table 25: Watchdog Timing (f_{CPU} = 8 MHz)

	CR Register Initial Value	WDG Timeout (ms)	Lock-up Timeout (µs)
Maximum	FFh	400	32
Minimum	C0h	6.250	

12.5 Interrupts

None.

12.6 Register Description

Table 26: Watchdog Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
001Bh	7F	R/W	WDGCR	WDGA	T[6:0]						

WDG CONTROL REGISTER (WDGCR)

Read/Write
Reset Value: 2 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = WDGA *Activation bit.*
This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0 Watchdog disabled

1 Watchdog enabled

Bits [6:0] = T[6:0] *7-bit Timer (MSB to LSB).*
These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 is cleared).

13 8-bit Timer (TIMA)

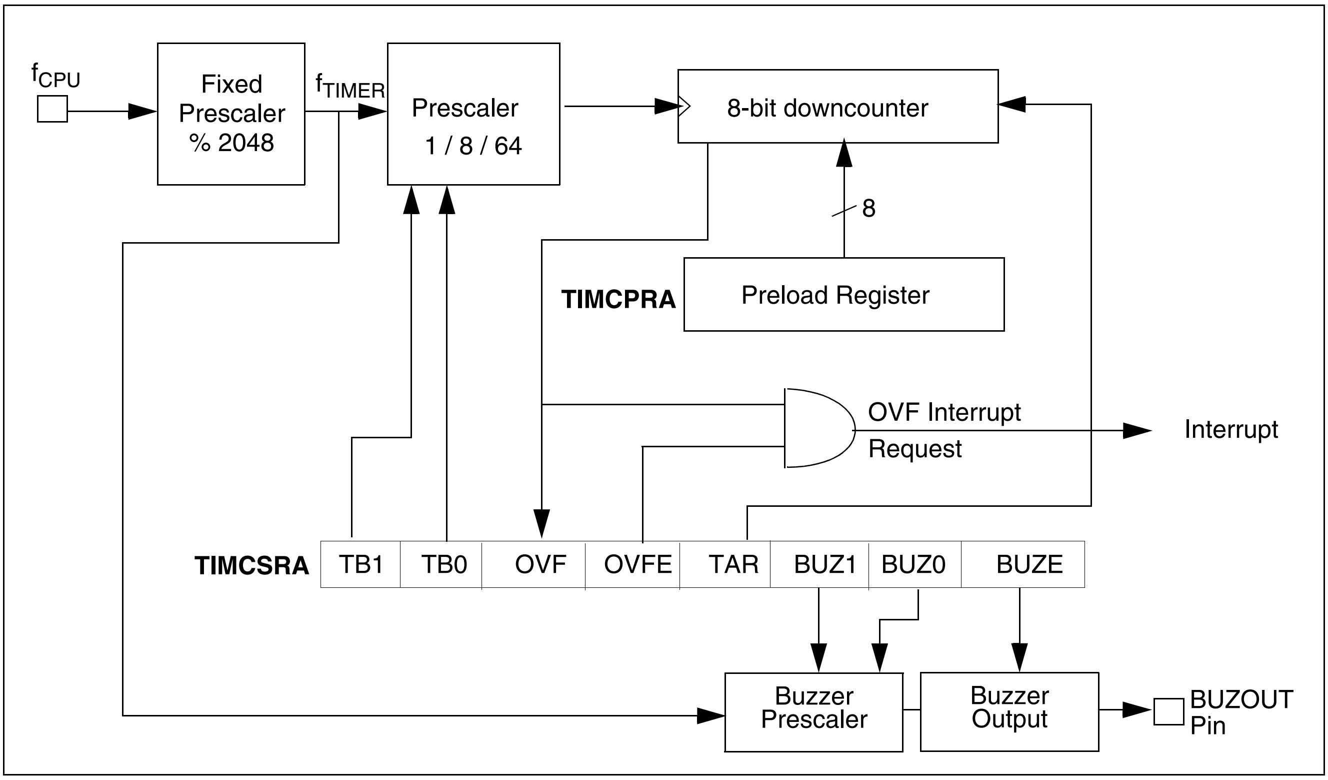
13.1 Introduction

Timer A is an 8-bit programmable free-running downcounter driven by a programmable prescaler. This block also has a buzzer. The block diagram is shown in Figure 41.

13.2 Main Features

- Programmable Prescaler: f_{CPU} divided by 1, 8 or 64.
- Overflow status flag and maskable interrupt
- Reduced power mode
- Independent buzzer output with 4 programmable tones

Figure 41: Timer A (TIMA) Block Diagram



13.3 Functional Description

Timer A is a 8-bit downcounter and its associated 8-bit register is loaded as start value of the downcounter each time it has reached the 00h value. A flag indicates that the downcounter rolled over the 00h value. The buzzer has 4 distinct tones. Before the downcounter prescaler block, the frequency is divided by 2048.

$$f_{TIMER} = f_{CPU}/2048$$

Note: In One-shot mode, the counter stops at 00h (low power state).

13.4 Register Description

Table 27: Timer Controller Register Map

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000Dh	00h	R/W	TIMCSRA	TB1	TB0	OVF	OVFE	TAR	BUZ1	BUZ0	BUZE
000Eh	00h	R/W	TIMCPRA	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

TIMER A CONTROL STATUS REGISTER (TIMCSRA)

Read/Write
Reset Value: (00h)

7	6	5	4	3	2	1	0
TB1	TB0	OVF	OVFE	TAR	BUZ1	BUZ0	BUZE

Bits [7:6] = TB[1:0] *Time Base period selection*
These bits are set and cleared by software.

00 Time base period = t_{TIMER} (256 µs @ 8 MHz)

01 Time base period = t_{TIMER} x 8 (2048 µs @ 8 MHz)

10 Time base period = t_{TIMER} x 64 (16384 µs @ 8 MHz)

11 Reserved

Bit 5 = OVF *Timer Overflow Flag.*
This bit is set by hardware. An interrupt is generated if OVFE = 1. It must be cleared by reading the TIMCSRA register.

0 No timer overflow.

1 The free-running downcounter reached 00h.

Bit 4 = OVFE *Timer Overflow Interrupt Enable.*
This bit is set and cleared by software.

0 Interrupt disabled

1 Interrupt enabled

Bit 3 = TAR Timer Auto-Reload
This bit is set and cleared by software.

0 One-shot mode. The counter restarts after a write in the TIMCPRA register.

1 Auto-Reload mode. The counter is reloaded automatically by the TIMCPRA register after the downcounter reaches 00h.

Bits [2:1] = BUZ[1:0] Buzzer tone selection
These bits are set and cleared by software.

00 Time base frequency = f_{TIMER}/16 (244 Hz @ 8 MHz)

01 Time base frequency = f_{TIMER}/8 (488 Hz @ 8 MHz)

10 Time base frequency = f_{TIMER}/4 (976 Hz@ 8 MHz)

11 Time base frequency = f_{TIMER}/2 (1.95 kHz @ 8 MHz)

Bit 0 = BUZE *Buzzer enable*
This bit is set and cleared by software.

0 Buzzer disabled

1 Buzzer enabled. It has priority over any other alternate function mapped onto the same pin (PWM).

TIMER A COUNTER PRELOAD REGISTER (TIMCPRA)

Read/Write
Reset Value: (00h)

7	6	5	4	3	2	1	0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

Bits [7:0] = PR[7:0] *Counter Preload Data*
These bits are set and cleared by software.

They are used to hold the reload value which is immediately loaded in the counter.

Note: The N number loaded in TIMCPRA register corresponds to a time of (N + 1) x Period timer.

The "00" value is prohobited.

14 8-bit Timer with External Trigger (TIMB)

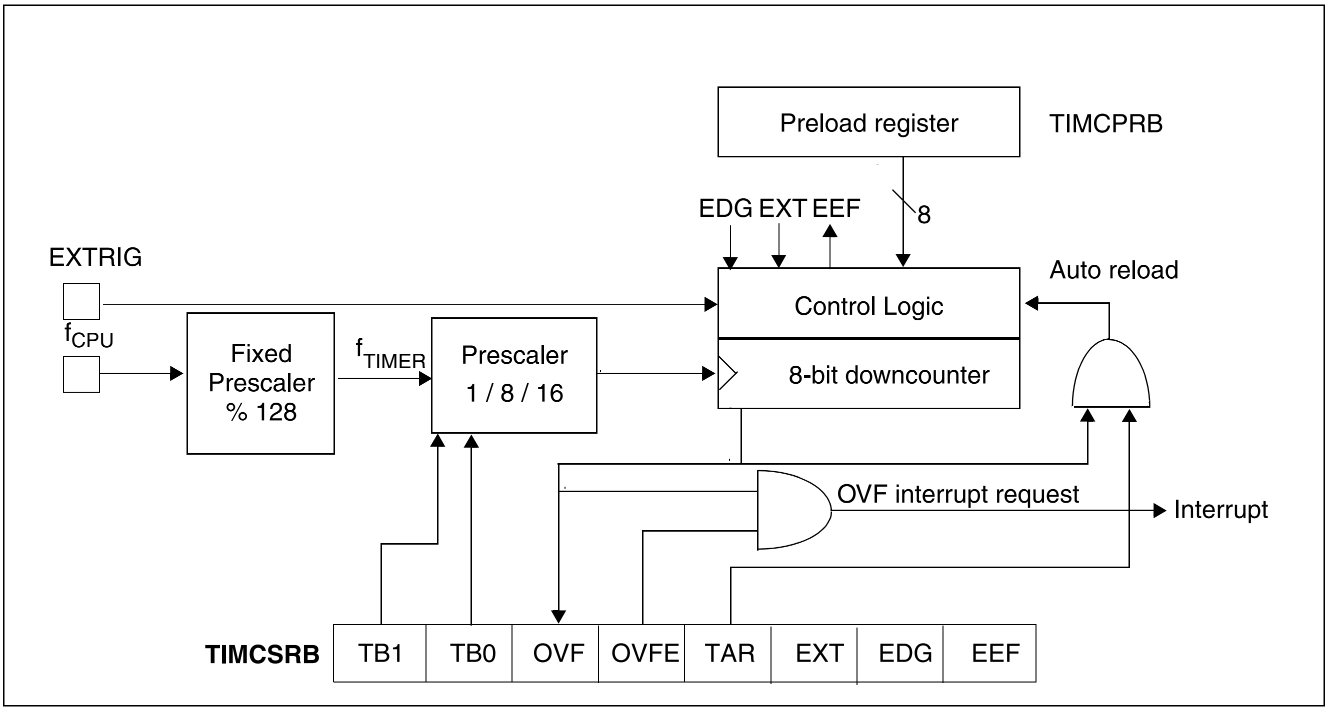
14.1 Introduction

Timer B is an 8 bit-programmable free-running downcounter, driven by a programmable prescaler. An external signal can also trigger the countdown. The Timer B block diagram is shown in Figure 42.

14.2 Main Features

- Programmable Prescaler: f_{CPU} divided by 1, 8 or 16
- Overflow status flag and maskable interrupt
- Auto reload capability
- An external signal with programmable polarity can trigger the count-down

Figure 42: External Timer Block Diagram



14.3 Functional Description

The 8 bit-downcounter timer counts from a start value down to 00h. The start value is preloaded from the associated 8-bit TIMCPRB register every time it is written, or when the counter has reached the 00h value (Auto Reload feature) if the TAR bit is set. The OVF flag is set when the downcounter reaches 00h. An interrupt is generated if the OVFE bit is set.

When the EXT bit is set, an external signal edge triggers the countdown start. The EDG bit controls the rising or falling signal edge. Once detected, the selected edge sets the EEF flag, preloads the downcounter with the start value and starts the countdown as usual.

During the countdown, the downcounter cannot be retriggered and subsequent pulses occurring after the countdown has started are ignored until the counter reaches 00h.

The four possible operating modes are described in Table 28.

Table 28: Timer Operating Mode

TAR	EXT	Timer mode
0	0	**One-shot** after the TIMCPRB register write (no auto reload)
0	1	**One-shot** after the external signal detection (no auto reload). Only the very first external pulse triggers the countdown (Note 2)
1	0	Downcounter **auto-reload** when 00h reached Downcounter reloaded with TIMCPRB register value, count-down restarts
1	1	**One-shot** for each external signal detection. Downcounter preloaded with TIMCPRB when 00h reached. Countdown restarts after the next external signal detection.

Note:1. The downcounter value cannot be read.

2. Change the EXT value to exit the External One-shot mode.

Table 29: Timer Controller Register Map

Address	Reset	R/W	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0038h	00h	R/W	TIMCSRB	TB1	TB0	OVF	OVFE	TAR	EXT	EDG	EEF
0039h	01h	R/W	TIMCPRB	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

TIMER B CONTROL STATUS REGISTER (TIMCSRB)

Read/Write
Reset value: (00h)

7							0
TB1	TB0	OVF	OVFE	TAR	EXT	EDG	EEF

Bits [7:6] = TB[1:0] *Time Base period selection*

These bits are set and cleared by software.

00 Time base period = t_{TIMER} (16 µs @ 8 MHz)

01 Time base period = t_{TIMER} x 8 (128 µs @ 8 MHz)

10 Time base period = t_{TIMER} x 16 (256 µs @ 8 MHz)

11 Reserved

Bit 5 = OVF *Timer Overflow Flag*

This bit is set by hardware. An interrupt is generated if OVFE = 1. It must be cleared by reading the TIMCSRB register.

0 No timer overflow

1 The free running downcounter rolled over from 00h

Bit 4 = OVFE *Timer Overflow Interrupt Enable*
This bit is set and cleared by software.

0 Interrupt disabled

1 Interrupt enabled

Bit 3 = TAR *Timer Auto Reload*

This bit is set and cleared by software.

0 One-shot mode. The counter restarts after writing to the TIMCPRB register.

1 Auto reload mode. The counter is reloaded automatically from the TIMCPRB register when 00h is reached.

Bit 2 = EXT *External Trigger*

This bit is set and cleared by software.

0 Internal. The downcounter restarts after writing to the TIMCPRB register or after an auto-reload if the TAR bit is set

1 External. The downcounter is preloaded with the TIMCPRB register but the countdown starts only when the external signal is detected, not by writng to the TIMCPRB register.

Bit 1 = EDG *External Signal Edge*

This bit is set and cleared by software.

0 A rising edge signal starts the count-down.

1 A falling edge signal starts the count-down

Bit 0 = EEF *External Event Flag*

This bit is set and cleared by hardware when an external event occurs.

This bit is cleared when the counter reaches “00h” in External mode or when the value of the EXT bit is changed by software.

In Internal mode, this bit is set when the selected edge is detected (the EDG bit) but it is never cleared by itself. It may then be used as a simple edge detector.

TIMER B COUNTER PRELOAD REGISTER (TIMCPRB)

Read/Write
Reset value: (01h)

7							0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

Bits [7:0] = PR[7:0] *Counter Preload Data*

This bit is set and cleared by software.

Bits hold the reload value which is loaded in the counter either immediately (EXT = 0) or when the external signal is detected (EXT = 1).

Note: *The N number loaded in TIMCPRB register corresponds to a time of (N + 1) x Period timer.*

The "00" value is prohibited.

15 Infrared Preprocessor (IFR)

The Infrared Preprocessor measures the intervals between 2 adjacent edges of a serial input.

15.1 Main Features

- Interval measurement between 2 edges (Time Base = 12.5 kHz) @ f_{CPU} = 8 MHz
- Choice of active edge
- Glitch filter
- Overflow detection (20.4 ms = 255/12.5 kHz)
- Maskable interrupt

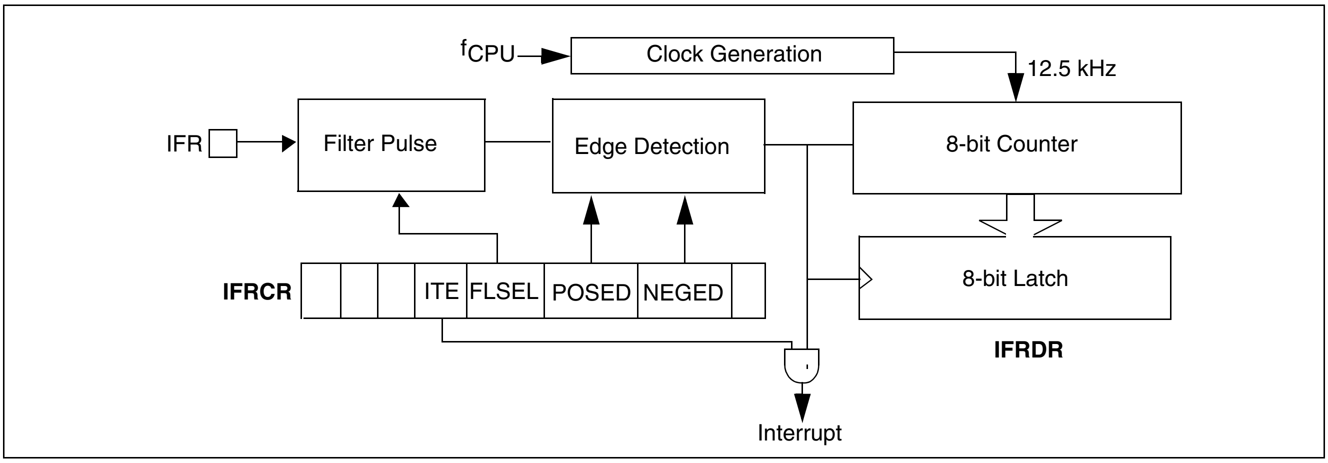
15.2 Functional Description

The IR Preprocessor measures the interval between two adjacent edges of the IFR input signal.

The POSED and NEGED bits determine if the intervals of interest involve:

- consecutive positive edges,
- negative edges,
- or any pair of edges as described in Table 30.

Figure 43: IFR Block Diagram



The measurement is a count resulting from a 12.5 kHz clock. Therefore, any pulse width that is less than 80 µs cannot be detected.

Whenever an edge of the specified polarity is detected, the count accumulated since the previously detected edge is latched into the IFRDR register, an interrupt is generated and the counter is reset.

If an edge is not detected within 20.4 ms (f_{CPU} = 8 MHz) and the count reaches its maximum value of 255, it is latched immediately. The internal interrupt flag and also an internal overflow flag are set. The latch content remains unchanged as long as the overflow flag is set.

The count stored in the latch register is overwritten in case the microcontroller fails to execute the read before the next edge. Writing to the IFRDR register clears the interrupt and internal overflow flag.

The IFR input signal is preprocessed by a spike filter. This filter removes all pulses with a positive level that lasts less than 2 µs or 160 µs, depending on the FLSEL bit. The negative level can be of any duration and is never filtered out.

Note: *If the interrupt is enabled but no signal is detected, an interrupt occurs every 20.4 ms.*

15.3 Register Description

INFRA RED DATA REGISTER (IFRDR)

Read/Write
Reset Value: (00h)

7	6	5	4	3	2	1	0
IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0

Bits [7:0] = IR[7:0] *Infra red pulse width*
The 8-bit counter value is transferred in this register when an expected edge occurs on the IFR pin or when the counter overflows. A write to this register resets the internal overflow flag.

INFRA RED CONTROL REGISTER (IFRCR)

Read/Write
Reset Value: (00h)

7	6	5	4	3	2	1	0
0	0	0	ITE	FLSEL	POSED	NEGED	0

Bits [7:5] = Reserved. Forced by hardware to 0.

Bit 4 = ITE *Interrupt enable*

0 Interrupt disabled

1 Interrupt enabled. It is generated when an edge (falling and/or rising depending on bits POSED and NEGED) occurs or after a counter overflow.

Bit 3 = FLSEL *Spike filter pulse width selection*

0 Filter positive pulses narrower than 2 µs

1 Filter positive pulses narrower than 160 µs

Bits [2:1] = POSED, NEGED *Edge selection for the duration measurement*

Table 30: Duration Measurement

POSED	NEGED	Count latch at...
0	0	When count reaches 255
0	1	Negative transition of IFR or when count reaches 255
1	0	Positive transition of IFR or when count reaches 255
1	1	Positive or negative transition of IFR or when count reached 255

Bit 0 = Reserved. Forced by hardware to 0.

16 Registers

16.1 Register Description

NAME REGISTER (NAMER)

Read only
Reset value: 00h

7	6	5	4	3	2	1	0
N[7:0]							

Bits [7:0] = N[7:0] *Circuit Name*
This register indicates the version number of the circuit. The current value is 01h.

Table 31: ST7FLCD1 Register Summary (Sheet 1 of 2)

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0000h	00h	R	NAMER								
0001h	00h	R/W	MISCR	0	0	0	0	0	PA5OVD	PA4OVD	0
0002h	00h	R/W	PADR	PADR[7:0]							
0003h	00h	R/W	PADDR	PADDR[7:0]							
0004h	00h	R/W	PBDR	PBDR[7:0]							
0005h	00h	R/W	PBDDR	PBDDR[7:0]							
0006h	00h	R/W	PCDR	PCDR[7:0]							
0007h	00h	R/W	PCDDR	PCDDR[7:0]							
0008h	00h	R/W	PDDR	PDDR[7:0]							
0009h	00h	R/W	PDDDR	PDDDR[7:0]							
000Ah	00h	R	ADCDR	AD[7:0]							
000Bh	00h	R/W	ADCCSR	COCO	0	ADON	0	0	0	CH[1:0]	
000Ch	00h	R/W	ITRFRE	0	0	ITB EDGE	ITBLAT	ITBITE	ITA EDGE	ITALAT	ITAITE
000Dh	00h	R/W	TIMCSRA	TB1	TB0	OVF	OVFE	TAR	BUZ1	BUZ0	BUZE
000Eh	00h	R/W	TIMCPRA	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
000Fh	00h	R/W	PWMDCR0	DCR0[7:0]							
0010h	00h	R/W	PWMDCR1	DCR1[7:0]							
0011h	00h	R/W	PWMDCR2	DCR2[7:0]							
0012h	00h	R/W	PWMDCR3	DCR3[7:0]							
0013h	00h	R/W	PWMCRA	OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0
0014h	FFh	R/W	PWMARRA	ARRA[7:0]							
0015h	00h	R/W	PWMDCR4	DCR4[7:0]							
0016h	00h	R/W	PWMDCR5	DCR5[7:0]							
0017h	00h	R/W	PWMCRB	0	0	OE5	OE4	0	0	OP5	OP4
0018h	FFh	R/W	PWMARRB	ARRB[7:0]							
0019h	00h	R/W	FCSR								

Table 31: ST7FLCD1 Register Summary (Sheet 2 of 2)

Address	Reset		Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
001Ah			Reserved								
001Bh	7F	R/W	WDGCR	WDGA	T[6:0]						
001Ch	00h	R/W	I2CCR	00		PE	0	START	ACK	STOP	ITE
001Dh	00h	R	I2CSR	EVF	AF	TRA	0	BTF	0	M/IDL	SB
001Eh	00h	R/W	I2CCCR	FM/SM	Filteroff	CC[5:0]					
001Fh	00h	R/W	I2CDR	DR[7:0]							
0020h	00h	R/W	DDCCRA	0	0	PE	DDCCIEN	0	ACK	STOP	ITE
0021h	00h	R	DDCSR1A	EVF	0	TRA	BUSY	BTF	ADSL	0	0
0022h	00h	R	DDCSR2A	0	0	0	AF	STOPF	0	BERR	DDCCIF
0023h	00h	R/W	DDCOAR1A	ADD[7:1]							0
0024h	00h	R/W	DDCOAR2A	ADD[7:1]							0
0025h	00h	R/W	DDCDRA	DR[7:0]							
0026h			Reserved								
0027h	00h	R/W	DDCDCRA	0	0	ENDCF	ENDCE	EDF	EDE	WP	DDC2BPE
0028h	00h	R/W	DDCCRB	0	0	PE	DDCCIEN	0	ACK	STOP	ITE
0029h	00h	R	DDCSR1B	EVF	0	TRA	BUSY	BTF	ADSL	0	0
002Ah	00h	R	DDCSR2B	0	0	0	AF	STOPF	0	BERR	DDCCIF
002Bh	00h	R/W	DDCOAR1B	ADD[7:1]							0
002Ch	00h	R/W	DDCOAR2B	ADD[7:1]							0
002Dh	00h	R/W	DDCDRB	DR[7:0]							
002Eh			Reserved								
002Fh	00h	R/W	DDCDCRB	0	0	ENDCF	ENDCE	EDF	EDE	WP	DDC2BPE
0030h	00h	R/W	DMCR	WDGOFF	MTR	BC2	BC1	BC0	BIR	BIW	AIE
0031h	10h	R	DMSR	WP	STE	STF	RST	BRW	BK2F	BK1F	AF
0032h	FFh	R/W	DMBK1H	BK1H7	BK1H6	BK1H5	BK1H4	BK1H3	BK1H2	BK1H1	BK1H0
0033h	FFh	R/W	DMBK1L	BK1L7	BK1L6	BK1L5	BK1L4	BK1L3	BK1L2	BK1L1	BK1L0
0034h	FFh	R/W	DMBK2H	BK2H7	BK2H6	BK2H5	BK2H4	BK2H3	BK2H2	BK2H1	BK2H0
0035h	FFh	R/W	DMBK2L	BK2L7	BK2L6	BK2L5	BK2L4	BK2L3	BK22L	BK2L1	BK2L0
0036h	00h	R/W	IFRDR	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
0037h	00h	R/W	IFRCR	0	0	0	ITE	FLSEL	POSED	NEGED	0
0038h	00h	R/W	TIMCSRB	TB1	TB0	OVF	OVFE	TAR	EXT	EDG	EEF
0039h	01h	R/W	TIMCPRB	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
003Ah			Reserved								

17 Electrical Characteristics

The ST7FLCD1 device contains circuitry to protect the inputs against damage due to high static voltage or electric field. Nevertheless it is advised to take normal precautions and to avoid applying to this high impedance voltage circuit any voltage higher than the maximum rated voltages. It is recommended for proper operation that V_{IN} and V_{OUT} be constrained to the range:

$$V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{DD}$$

To enhance reliability of operation, it is recommended to connect unused inputs to an appropriate logic voltage level such as V_{SS} or V_{DD}. All the voltages in the following table, are referenced to V_{SS}.

17.1 Absolute Maximum Ratings

Table 32: Absolute Maximum Ratings

Symbol	Ratings	Value	Unit
V_{DD}	Recommended Supply Voltage	-0.3 to +6.0	V
V_{IN}	Input Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
V_{AIN}	Analog Input Voltage (A/D Converter)	V_{SS} -0.3 to V_{DD} + 0.3	V
V_{OUT}	Output Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
I_{IN}	Input Current	-10 to +10	mA
I_{OUT}	Output Current	-10 to +10	mA
I_{INJ}	Accumulated injected current of all I/O pins (V_{DD}, V_{SS})	40	mA
T_A	Operating Temperature Range	0 to +70	°C
T_{STG}	Storage Temperature Range	-65 to +150	°C
T_J	Junction Temperature	150	°C
PD	Power Dissipation	TBD	mW
ESD	ESD susceptibility	2000	V

17.2 Power Considerations

The average chip-junction temperature, T_J, in degrees Celsius, may be calculated using the following equation:

$$T_J = T_A + (P_D \times \theta J_A) \ (1)$$

Where:

- T_A is the Ambient Temperature in °C,
- θJ_A is the Package Junction-to-Ambient Thermal Resistance, in °C/W,
- P_D is the sum of P_{INT} and $P_{I/O}$,
- P_{INT} is the product of I_{DD} and V_{DD}, expressed in Watts. This is the Chip Internal Power
- $P_{I/O}$ represents the Power Dissipation on Input and Output Pins; User Determined.

For most applications $P_{I/O}<P_{INT}$ and may be neglected. $P_{I/O}$ may be significant if the device is configured to drive Darlington bases or sink LED Loads. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is given by:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Therefore:

$$K = P_D \times (T_A + 273°C) + \theta J_A \times P_D^2 \quad (3)$$

Where:

- K is a constant for the particular part, which may be determined from equation (3) by measuring P_D (at equilibrium) for a known $T_{A.}$ Using this value of K, the values of P_D and T_J ay be obtained by solving equations (1) and (2) iteratively for any value of T_A.

17.3 Thermal Characteristics

Table 33: Thermal Characteristics

Symbol	Package	Value	Unit
θJ_A	28-pin Small Outline Packqge (SO28)	69	°C/W

17.4 AC/DC Electrical Characteristics

All voltages are referred to VSS and T_A = 0 to +70°C (unless otherwise specified)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
General						
V_{DD}	Operating Supply Voltage		4.5	5	5.5	V
	Operating Voltage for FLASH access	READ	3.8			V
		WRITE/ERASE	4.5		5.5	V
I_{DD}	CPU RUN mode	I/O in input mode V_{DD} = 5V f_{CPU} = 8 MHz, T_A = 20° C		14	18	mA
	CPU WAIT mode			12	18	mA
	CPU HALT mode			1	10	uA
Control Timing						
f_{OSC}	External frequency			24	27	MHz
f_{CPU}	Internal frequency			8	9	MHz
t_{BU}	Startup Time Built-Up Time	Crystal Resonator		8	20	ms
t_{RL}	External RESET Input pulse Width		1000			ns
t_{PORL}	Internal Power Reset Duration		4096			t_{CPU}
t_{POWL}	Watchdog RESET Output Pulse Width		500			ns
t_{DOG}	Watchdog Time-out	f_{CPU} = 8 MHz	50000 6.25		3200000 400	t_{CPU} ms

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{ILIL}	Interrupt Pulse Period			See Note 1		t_{CPU}
t_{OXOV}	Crystal Oscillator Start-up Time				50	ms
t_{DDR}	Power-up rise time	V_{DD} min.	1		100	ms
Standard I/O Port Pins						
V_{OL}	Output Low Level Voltage Port A[7:6,3:0], Port B[3:0], Push Pull	I_{OL} = 2 mA and V_{DD} = 5 V			0.4	V
V_{OL}	Output Low Level Voltage Port C[1:0] Open Drain	I_{OL} = 4 mA and V_{DD} = 5 V			0.4	V
V_{OL}	Output Low Level Voltage Port A[5:4] (See Note 2)	I_{OL} = 8 mA and V_{DD} = 5 V I_{OL} = 2 mA and V_{DD} = 5 V			0.4	V
V_{OL}	Output Low Level Voltage Port D[7:0] Open Drain	I_{OL} = 4 mA and V_{DD} = 5 V			0.4	V
V_{OH}	Output High Level Voltage Port A[7:6,3:0], Port B[3:0], Push Pull	I_{OH} = 2 mA	V_{DD}-0.8			V
V_{OH}	Output High Level Voltage Port A[5:4]	I_{OH} = 2 mA I_{OH} = 8 mA	V_{DD}-0.8			V
V_{OH}	Output High Level Voltage Port C[1:0], Port D (See Note 3)				V_{DD}	V
V_{IH}	Input High Level Voltage Port A [7:0], Port B [3:0], Port C [1:0], Port D[7:0], $\overline{RESET}$	Leading Edge	0.7 * V_{DD}		V_{DD}	V
V_{IL}	Input Low Level Voltage Port A [7:0], Port B [3:0], Port C [1:0], Port D[7:0], $\overline{RESET}$	Trailing Edge	V_{SS}		0.2 * V_{DD}	V
I_{IL}	I/O Ports Hi-Z Leakage Current Port A[7:0], Port B[3:0], Port C[1:0], Port D[7:0], $\overline{RESET}$				10	µA
C_{OUT} C_{IN}	Capacitance: Ports (as Input or Output), $\overline{RESET}$				12 8	pF
IRPU	Pull-up resistor current	V_{DD} = 5V V_{IN} = V_{SS} T = 25°C	30	60	100	µA

Note:1. The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 cycles.

2. For the case of I_{OL} = 8 mA, 8 mA output current if corresponding overdrive bit = 1 in MISCR register.

3. Output high level by means of external pull-up resistor.

17.5 Power On/Off Electrical Specifications

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{TRH}	Power ON/OFF Reset Trigger V_{DD} rising edge	V_{DD} Variation 50mV/mS	3.8	4	4.2	V
V_{TRL}	Power ON/OFF Reset Trigger V_{DD} falling edge	V_{DD} Variation 50mV/mS	3.75	4	4.2	V
V_{TRM}	V_{DD} minimum for Power ON/OFF Reset active	V_{DD} Variation 50mV/mS		TBD		V

17.6 8-bit Analog-to-Digital Converter

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
f_{ADC}	Analog control frequency	V_{DD} = 5 V			2	MHz
|TUE|	Total unadjusted error	f_{CPU} = 8 MHz, f_{ADC} = 2MHz V_{DD} = 5 V	0	1	2	LSB
OE	Offset error		-2	1	2	
GE	Gain error		-2	1	2	
|DLE|	Differential linearity error		0	0.5	1	
|ILE|	Integral linearity error		0	1	2	
V_{AIN}	Conversion range voltage		V_{SS}		V_{DD}	V
I_{ADC}	A/D conversion supply current	f_{CPU} = 8 MHz, f_{ADC} = 2MHz V_{DD} = 5 V		1		mA
t_{STAB}	Stabilization time after enable ADC				1	µs
t_{LOAD}	Sample capacitor loading time			1 4		µs $1/f_{ADC}$
t_{CONV}	Conversion time			2 8		µs $1/f_{ADC}$
R_{AIN}	External input resistor				15	kΩ
R_{ADC}	Internal input resistor			1.5		kΩ
C_{SAMPLE}	Sample capacitor			6		pF

17.7 I2C/DDC Bus Electrical Specifications

Symbol	Parameter	Standard mode		Fast mode		Unit
		Min.	Max.	Min.	Max.	
V_{HYS}	Hysteresis of Schmitt trigger inputs Fixed input levels V_{DD}-related input levels	 na na	 na na	 0.2 0.05 V_{DD}		V
T_{SP}	Pulse width of spikes which must be suppressed by the input filter	na	na	0 ns	50 ns	ns
T_{OF}	Output fall time from VIH min to VIL max with a bus capacitance from 10 pF to 400 pF with up to 3 mA sink current at VOL1 with up to 6 mA sink current at VOL2	 na	 250 na	 20+0.1Cb 20+0.1Cb	 250 250	ns
I	Input current each I/O pin with an input voltage between 0.4V and 0.9 V_{DD} max	- 10	10	-10	10	µA
C	Capacitance for each I/O pin		10		10	pF

Note: na = not applicable
Cb = capacitance of one bus in pF

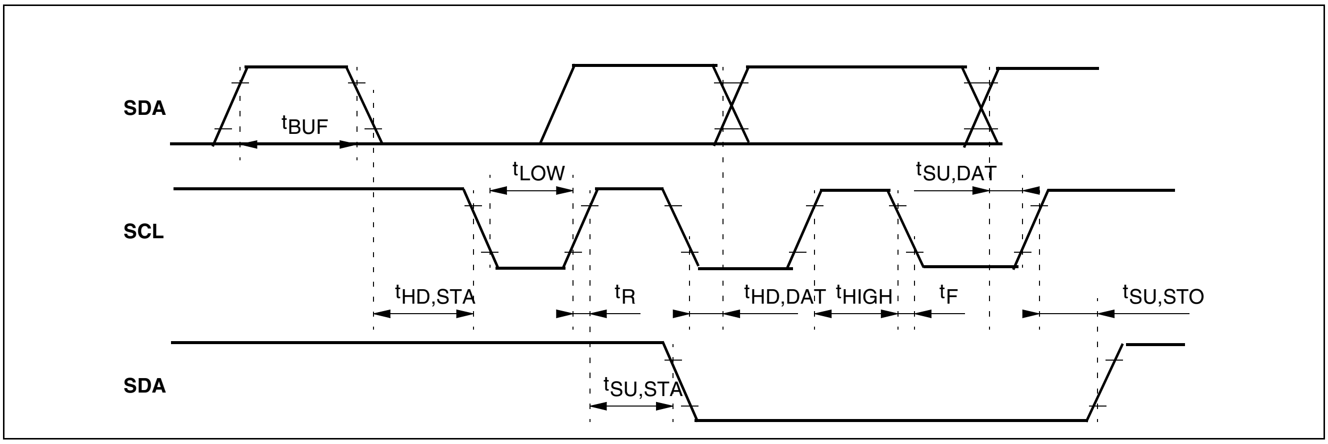
17.8 I2C/DDC Bus Timings

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
t_F	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

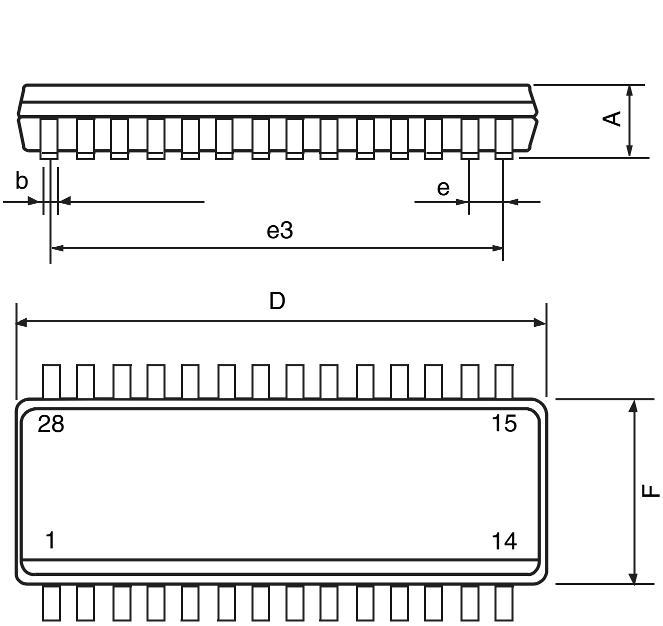
Note:1. The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.

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.*
3. *Cb = total capacitance of one bus line in pF.*
4. *I²C parameters compliant with I²C Bus Specification for speeds up to 400 kHz only. Faster speeds are at user responsibility.*

Figure 44: I²C Bus Timing



18 Package Mechanical Data



Dimensions	Mm			Inch		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			2.65			0.104
a1	0.1		0.3	0.004		0.012
b	0.35		0.49	0.014		0.019
b1	0.23		0.32	0.009		0.013
c		0.5			0.020	
c1	45° (typ.)					
D	17.7		18.1	0.697		0.713
E	10		10.65	0.394		0.419
e		1.27			0.050	
e3		16.51			0.65	
F	7.4		7.6	0.291		0.299
L	0.4		1.27	0.016		0.050
S	8° (max.)					

19 Revision History

Table 34: Summary of Modifications

Date	Version	Description
May 2002	2.1	Addition of Section 5.5: In-Circuit Programming (ICP) and Section 5.6: In-Application Programming (IAP).
19 Aug 2002	2.2	Update of Chapter numbering system. Update of Figure 2: 28-pin Small Outline Package (SO28) Pinout, Figure 12: Typical ICP Interface, Pin 14 becomes PA6/ITA/EXTRIG. Addition of MISCR register (0001h) and update of DDC2B Control Register data. Lock-up Counter info added in Section 12: Watchdog Timer (WDG). Buzzer output info added in Section 13: 8-bit Timer (TIMA). Modification of oscillator frequency from 24 MHz to maximum of 27 MHz, Fast I²C mode up to 800 kHz (for certain applications), Section 8: PWM Generator, Section 10.5: Transfer Sequencing and Section 11.6: Transfer Sequencing. Addition of Section 17: Electrical Characteristics and Section 19: Revision History.
24 Sept 2002	2.3	Modification of Figure 4: Program Memory Map.
14 Oct 2002	2.4	Modification of Figure 4: Program Memory Map and Table 34: Summary of Modifications.
4 Dec 2002	2.5	Modification of I²C Clock Control register.
6 Feb 2003	2.6	Change of V_{TRH} and V_{TRL} values in Section 17.4: AC/DC Electrical Characteristics.
11 Feb 2003	2.7	Addition of Section 1.5: External Connections. Update of DDC2B Control Register (Bit 3) information in Section 11.7: Register Description. Change of V_{DD} values in Section 17.4: AC/DC Electrical Characteristics.
2 Sept 2003	2.8	Modification of values in Section 17.4: AC/DC Electrical Characteristics, Section 17.5: Power On/Off Electrical Specifications and Section 17.6: 8-bit Analog-to-Digital Converter.
13 April 2004	2.9	Addition of V_{OH} row in Standard I/O Port Pins on page 89. Addition of Note 1 on page 9.
9 Feb 2005	2.10	Update of Section 6.1.2: Crystal Oscillator Mode on page 32.