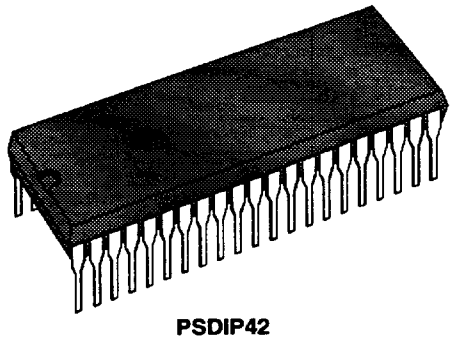
SGS-THOMSON MICROELECTRONICS

ST7271

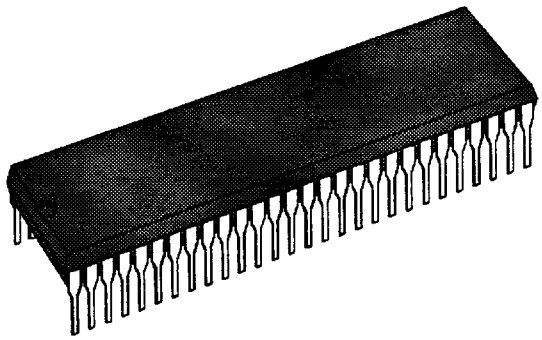
8-BIT HCMOS MCUs WITH EEPROM AND TV/MONITOR DEDICATED FUNCTIONS

PRELIMINARY DATA

- 5V ± 10% supply operating range
- 4MHz Maximum Internal Clock Frequency
- Fully static operation
- 0 to +70°C Operating Temperature Range
- Run, Wait, and Stop Modes
- User ROM: up to 15144 bytes
  Data RAM: up to 256 bytes
  EEPROM: up to 512 bytes
  EWPCC EEPROM: 256 bytes
- 56 pin Shrink Dual In Line Package (ST7271N)
  42 pin Shrink Dual In Line Package (ST7271J)
- up to 27 I/O lines
  8 I/O Open Drain with 12V capability
- up to 8 lines programmable as interrupt wake-up inputs
- 16-bit timer with 2 input capture and 2 output compare functions
- Sync Processor for video timing analysis
- East/West Pin Cushion Automatic Correction with DAC output.
- Watchdog for system reliability and integrity
- 8-bit Analog to Digital Converter with up to 8 channels
- 16 10-bit PWM/BRM Digital to Analog outputs
- 2 12-bit PWM/BRM Digital to Analog outputs
- Industry Standard Serial Peripheral Interface
- User mask options:
  - SPI Data Rate
  - Watchdog enable/disable after Reset
  - Watchdog enable during WAIT mode
- Master Reset and Power-on reset
- Full Hardware Emulator
- 8-bit data manipulation
- 74 basic instructions
- 10 main addressing modes
- 8x8 unsigned multiply instruction
- true bit manipulation
- Complete development support on Real-time emulator with PC/DOS
- Full software package (Cross Assembler, debugger)



PSDIP42



PSDIP56

(Ordering Information at the end of the Datasheet)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	PACKAGE
ST7271N5	16K	256	512	PSDIP56
ST7271N3	12K	256	512	PSDIP56
ST7271N1	8K	192	384	PSDIP56
ST7271J1	8K	192	384	PSDIP42

December 1993

Figure 1a. 56 Pin Shrink DIP Pinout

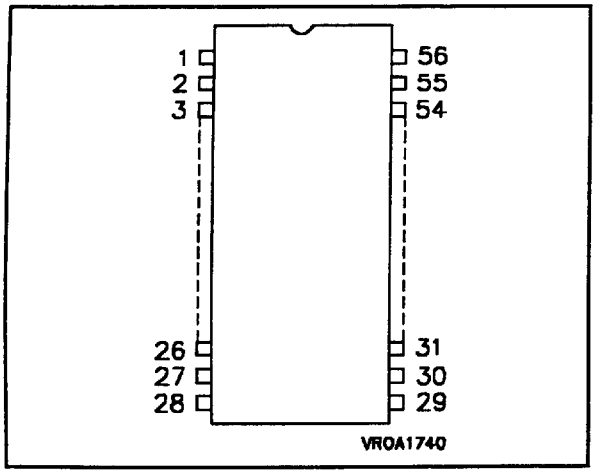
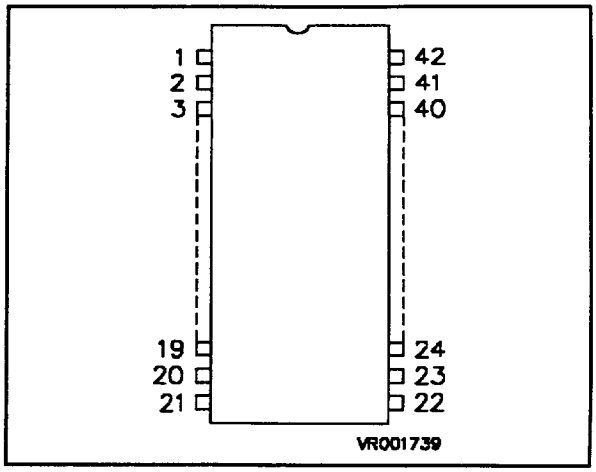


Figure 1b. 42 Pin Shrink DIP Pinout



ST7271N Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	56	V_{SSA}
2	EWPCC	55	V_{SS}
3	DA0	54	PC5/$\overline{SS}$
4	DA1	53	PC4/SCK
5	DA2	52	PC3/MOSI
6	DA3	51	PC2/MISO
7	DA4	50	PC1
8	DA5	49	PC0/OCMP
9	DA6	48	TEST
10	DA7	47	DA17
11	DA8	46	DA16
12	DA9	45	DA15
13	PB7	44	DA14
14	PB6	43	PA0
15	PB5	42	PA1
16	PB4	41	PA2
17	PB3	40	PA3
18	PB2	39	PA4
19	PB1	38	PA5
20	VFBACK/PB0	37	PA6
21	PD4	36	PA7
22	CLMPO/PD3	35	DA13
23	DA10	34	DA12
24	DA11	33	OSCIN
25	$\overline{RESET}$	32	OSCOUT
26	VSYNCO/PD2	31	CSYNCI/PDO
27	VSYNCI	30	HSYNCO/PD1
28	V_{DD}	29	HSYNCI

ST7271J Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	42	V_{SSA}
2	EWPCC	41	V_{SS}
3	DA0	40	PC5/$\overline{SS}$
4	DA1	39	PC4/SCK
5	DA2	38	PC3/MOSI
6	DA3	37	PC2/MISO
7	DA4	36	PC0/OCMP
8	DA5	35	TEST
9	DA6	34	PA0
10	DA7	33	PA1
11	DA8	32	PA2
12	DA9	31	PA3
13	PB3	30	PA4
14	PB2	29	PA5
15	PB1	28	PA6
16	VFBACK/PB0	27	PA7
17	CLMPO/PD3	26	OSCIN
18	$\overline{RESET}$	25	OSCOUT
19	VSYNCO/PD2	24	CSYNCI/PDO
20	VSYNCI	23	HSYNCO/PD1
21	V_{DD}	22	HSYNCI

1 GENERAL DESCRIPTION

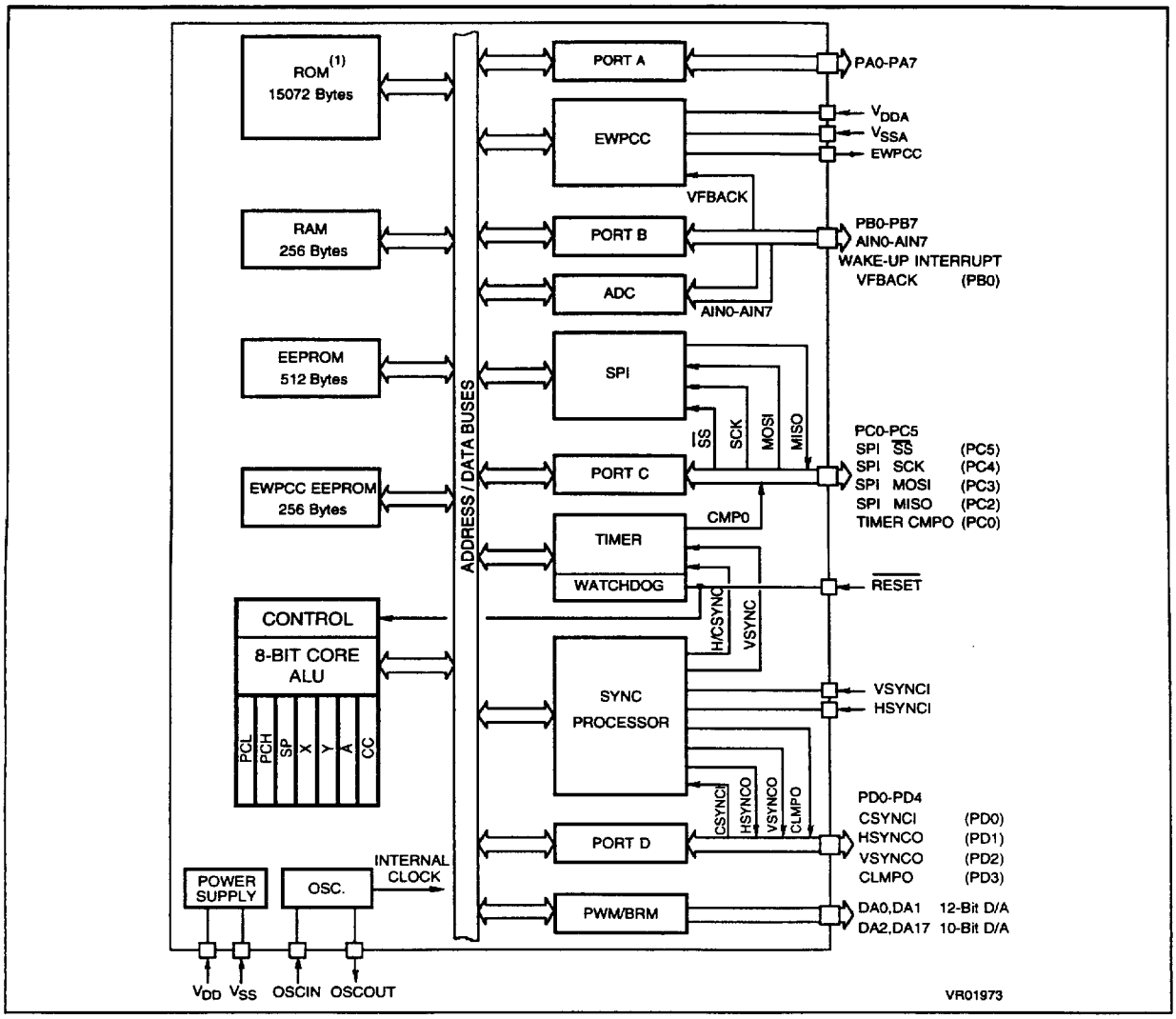
1.1 INTRODUCTION

The ST7271 is a HCMOS microcontroller unit (MCU) from the ST72 family with dedicated peripherals for TV and Monitor applications.
It is based around an industry standard 8-bit core and offers an enhanced instruction set. The processor runs with an external clock at 8 MHz with a 5V supply. Due to the fully static design of this device, operation down to DC is possible. Under software control the ST7271 can be placed in WAIT or STOP mode thus reducing power consumption. The enhanced instruction set and addressing modes afford real programming potential.
In addition to standard 8-bit data management the ST7271 features true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

The device includes an on-chip oscillator, CPU, ROM, RAM, EEPROM, I/O, a timer with 2 input capture and 1 output compare signals, an 8-channel Analog to Digital Converter and an industry standard SPI as standard peripherals.

Dedicated functions include a Sync Processor for video timing analysis, East-West Pin Cushion automatic correction and 18 PWM/BRM outputs for analog control of external functions.

Figure 2. ST7271 Block Diagram



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

1.2 PIN DESCRIPTION

V_{DD}. Power supply voltage

V_{SS}. Digital Ground

V_{DDA}. Analog V_{DD} and reference for EWPCC Digital to Analog Converter (DAC, 8 Volts).

V_{SSA}. Analog V_{SS} for EWPCC DAC.

OSCIN, OSCOUT. Oscillator input and output pins. These pins are to be connected to a parallel resonant crystal or ceramic resonator. An external clock source can also be input on OSCIN.

$\overline{\text{RESET}}$. The active low input signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog has triggered. It can be used to reset external peripherals.

TEST. This pin must be held low for normal operation

VFBACK (PB0). Vertical Flyback signal (TTL level). This pin accepts the Vertical Flyback signal used for timing correlation for the East-west Pin Cushion correction when this is used or is PB0.

EWPCC. Analog output of correction signal from East-West Pin Cushion controller (2-6V, I_{OUT} = 1mA).

OCMP (PC0). Output compare signal coming from the TIMER. This output signal, according to a register bit option, can be the OCMP pin (for output compare 1 of the timer) or the PC0 pin.

MISO (PC2). SPI Master Out/Slave In Data Output/Input when SPI is enabled or PC2.

MOSI (PC3). SPI Master In/Slave Out Data Input/Output when SPI is enabled or PC3.

SCK (PC4). SPI Serial Clock when SPI is enabled or PC4

$\overline{\text{SS}}$ (PC5). SPI Slave Select when SPI is enabled or PC5.

VSYNCI. Vertical Sychronization Input (TTL level)

HSYNCI. Horizontal Sychronization Input (TTL level)

CSYNCI (PD0). Composite Sychronization Input (TTL level). This pin accepts the composite synchronisation input when the Sync Processor I/O functions are enabled or is PD0.

HSYNCO (PD1). Horizontal Sychronization Output. This pin outputs the horizontal synchronisation output from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD1.

VSYNCO (PD2). Vertical Sychronization Output. This pin outputs the vertical synchronisation output from the Sync Processor (or VSYNCI) when the Sync Processor I/O functions are enabled or is PD2.

CLMPO (PD3). Clamp Output. This pin outputs the clamping (back porch) output signal from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD3.

DA2-DA17 (56-pin package),
DA2-DA9 (42-pin package), 10-bit PWM/BRM outputs (for Analog controls, after external filtering)

DA0, DA1. 12-bit PWM/BRM outputs (for Analog Controls, after external filtering).

PA0-PA7, PB0-PB7, PC0-PC5, PD0-PD4 (56 pin package). These 27 lines are standard I/O lines, programmable as either input or output.

- **PORT A.** 8 I/O lines, bit programmable, accessed through DDRA and DRA Registers. Each bit can be defined as a standard input port bit without pull-up resistor or as an open drain output port (up to 12V).

- **PORT B.** 8 Standard I/O lines bit programmable accessed through DDRB and DRB Registers. Each bit can be programmed as an analog input (by control bits in the PORT B Configuration register), digital input (with internal pull-up resistor), push-pull digital output or as interrupt wake-up (with pull-up). These negative edge or low-level sensitive interrupt lines can wake-up the ST7271 from WAIT or STOP mode. This feature allows to build low power applications when the ST7271 can be waken-up from keyboard push.
PB0 is used for the East-West Pin cushion controller VFBACK input as shown above when the EWPCC is used.

- **PORT C.** 6 Standard I/O lines accessed through DDRC and DRC Registers. Each bit can be programmed as digital input (with or without pull-up internal resistor), open drain output or SPI control and data signals (as shown for the dedicated SPI signals above). Whenever the SPI is active, the outputs are in the pull-pull configuration.
The pull-up resistor is enabled for all bits present by one control bit in the Programmable Input/Output Configuration Register. The resistor is automatically disabled for the pins used for the SPI when the SPI is enabled.

- **PORT D.** 4 Standard I/O lines bit programmable accessed through DDRD and DRD Registers. Each bit can be programmed as an input (with internal pull-up resistor), push-pull output or Synchronization inputs and outputs to/from the Sync Processor. When programmed as inputs, Video Synchronisation signals can be directly inspected. The inputs may also be passed through the Sync Processor to the Timer Input Captures

These pin functions are also summarised in the following table, which also indicates the availability of functions for the 42-pin SDIP package.

PIN DESCRIPTION (Continued)

Table 1. ST7271 Pin Description

Pin Name	Pin Function(s)	56 Pins	42 Pins
V_{DDA}	Analog V_{DD} for EWPCC	1	1
EWPCC	EWPCC output voltage	2	2
DA0	12-bit PWM/BRM output*	3	3
DA1	12-bit PWM/BRM output	4	4
DA2	10-bit PWM/BRM output*	5	5
DA3	10-bit PWM/BRM output	6	6
DA4	10-bit PWM/BRM output	7	7
DA5	10-bit PWM/BRM output	8	8
DA6	10-bit PWM/BRM output	9	9
DA7	10-bit PWM/BRM output	10	10
DA8	10-bit PWM/BRM output	11	11
DA9	10-bit PWM/BRM output	12	12
PB7	I/O Port PB7	13	
PB6	I/O Port PB6	14	
PB5	I/O Port PB5	15	
PB4	I/O Port PB4	16	
PB3	I/O Port PB3	17	13
PB2	I/O Port PB2	18	14
PB1	I/O Port PB1	19	15
VFBACK/PB0	I/O Port PB0 /VFBACK Input	20	16
PD4	I/O Port PD4	21	
CLMPO/PD3	I/O Port PD3/Clamp Output	22	17
DA10	10-bit PWM/BRM output 10	23	
DA11	10-bit PWM/BRM output 11	24	
$\overline{\text{RESET}}$	Reset Input/Output	25	18
VSYNCO/PD2	I/O Port PD2/VSYNC Output	26	19
VSYNCI	VSYNC Input to Sync Processor	27	20
V_{DD}	Power Supply	28	21
HSYNCI	HSYNC Input to Sync Processor	29	22

Pin Name	Pin Function(s)	56 Pins	42 Pins
HSYNCO/PD1	I/O Port PD1/HSYNC Output	30	23
CSYNCI/PD0	I/O Port PD0/CSYNC Input	31	24
OSCOUT	Oscillator Output	32	25
OSCIN	Oscillator Input	33	26
DA12	10-bit PWM/BRM output 12	34	
DA13	10-bit PWM/BRM output 13	35	
PA7	I/O Port PA7	36	27
PA6	I/O Port PA6	37	28
PA5	I/O Port PA5	38	29
PA4	I/O Port PA4	39	30
PA3	I/O Port PA3	40	31
PA2	I/O Port PA2	41	32
PA1	I/O Port PA1	42	33
PA0	I/O Port PA0	43	34
DA14	10-bit PWM/BRM output 14	44	
DA15	10-bit PWM/BRM output 15	45	
DA16	10-bit PWM/BRM output 16	46	
DA17	10-bit PWM/BRM output 17	47	
TEST	TEST input, must be held to Vss	48	35
PC0/OCMP	I/O Port PC0, Timer Output Compare	49	36
PC1	I/O Port PC1	50	
PC2/MISO	I/O Port PC2, SPI Data	51	37
PC3/MOSI	I/O Port PC3, SPI Data	52	38
PC4/SCK	I/O Port PC4, SPI Clock output	53	39
PC5/$\overline{\text{SS}}$	I/O Port PC5, SPI Slave Select	54	40
Vss	Digital ground	55	41
V_{SSA}	Analog ground for EWPCC	56	42

Note * : Open Drain

1.3 CENTRAL PROCESSING UNIT

1.3.1 Introduction

The CPU has a full 8-bit parallel architecture. Six internal registers allow efficient 8-bit data manipulations. The CPU is able to execute 74 basic instructions with 9 main addressing modes. It is able to address 16k bytes of memory and registers with its program counter.

1.3.2 CPU Registers

The 6 CPU registers are shown in the programming model in Figure 3. Following an interrupt, all registers except Y are pushed onto the stack in the order shown in Figure 4. They are popped from stack in the reverse order.
The Y register is not affected by these automatic procedures. The interrupt routine must therefore handle Y, if needed, through the POP and PUSH instructions.

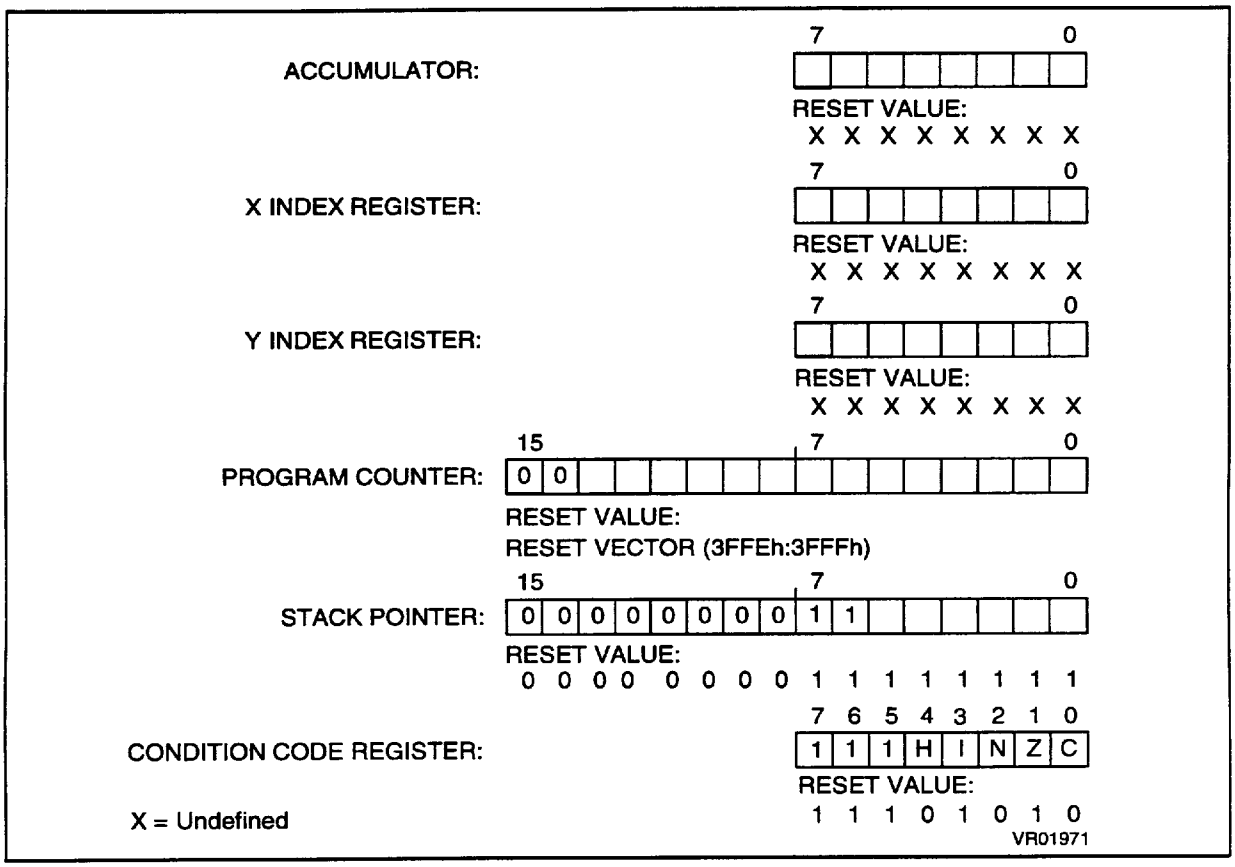
Accumulator (A). The accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations as well as data manipulations.

Index Registers (X and Y). These 8-bit registers are used to create effective addresses or as temporary storage area for data manipulations. The Y register is never automatically stacked. Interrupt routines must push or pop it by using the POP and PUSH instructions.

Program Counter (PC). The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. In the ST7271, only the 14 low order bits are used, bits 14 and 15 are forced to '0' giving an addressing range of 0- 3FFFh.

Stack Pointer (SP). The stack pointer is a 16-bit register. The 6 least significant bits contain the address of the next free location of the stack. The 10 most significant bits are forced as indicated in Figure 3.

Figure 3. Programming Model



The stack is used to save the CPU context on subroutines calls or interrupts. The user can also directly manipulate the stack through the PUSH and POP instructions.

After a MCU reset or after the reset stack pointer instruction (RSP), the stack pointer is set to its upper value (FFh). It is then decremented after data has been pushed onto the stack and incremented after data is popped from the stack. When the lower limit is exceeded, the stack pointer wraps around to the stack upper limit. The previously stored information is then overwritten and therefore lost.

A subroutine call occupies two stack locations and an interrupt five locations.

1.3.3 Condition Code Register (CC).

The condition code register is a 5 bit register which indicates the result of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. The following paragraphs describe each bit.

Half carry bit (H). The H bit is set to 1 when a carry occurs between the bits 3 and 4 of the ALU during an ADD, ACC, SUB or SBC instructions. The H bit is useful in BCD arithmetic subroutines.

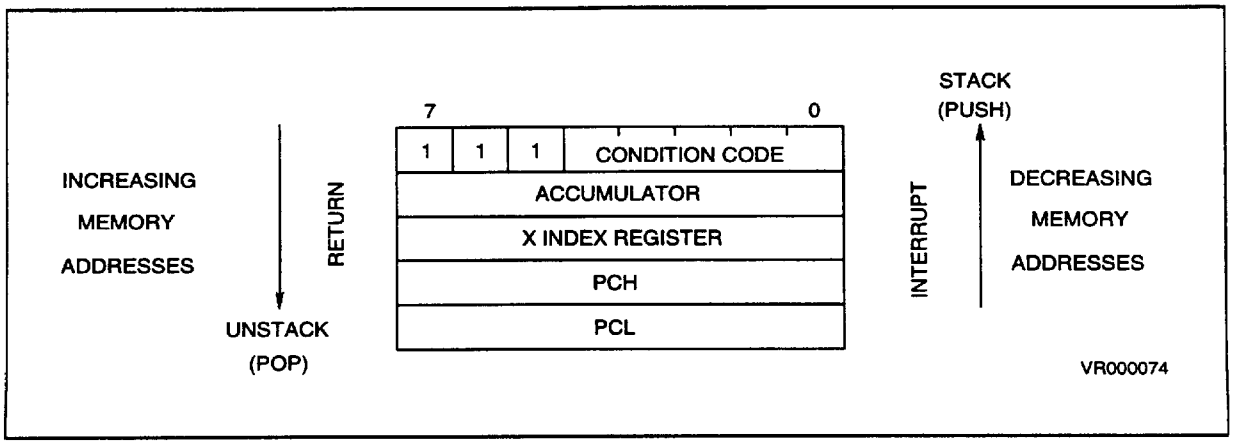
Interrupt mask (I). When the I bit is set to 1, all interrupts except the TRAP software interrupt are disabled. Clearing this bit enables interrupts to be passed to the core. Interrupts requested while I is set are latched and can be processed when I is cleared (only one interrupt request per interrupt enable flag can be latched).

Negative (N). When set to 1, this bit indicates that the result of the last data manipulation is negative (i.e. the most significant bit is a logic 1).

Zero (Z). When set to 1, this bit indicates that the result of the last data manipulation is zero.

Carry/Borrow (C). When set, C indicates that a carry or borrow out of the ALU occurred during the last arithmetic operation. This bit is also affected during bit test and branch, shift and rotate instructions.

Figure 4. Stacking Order



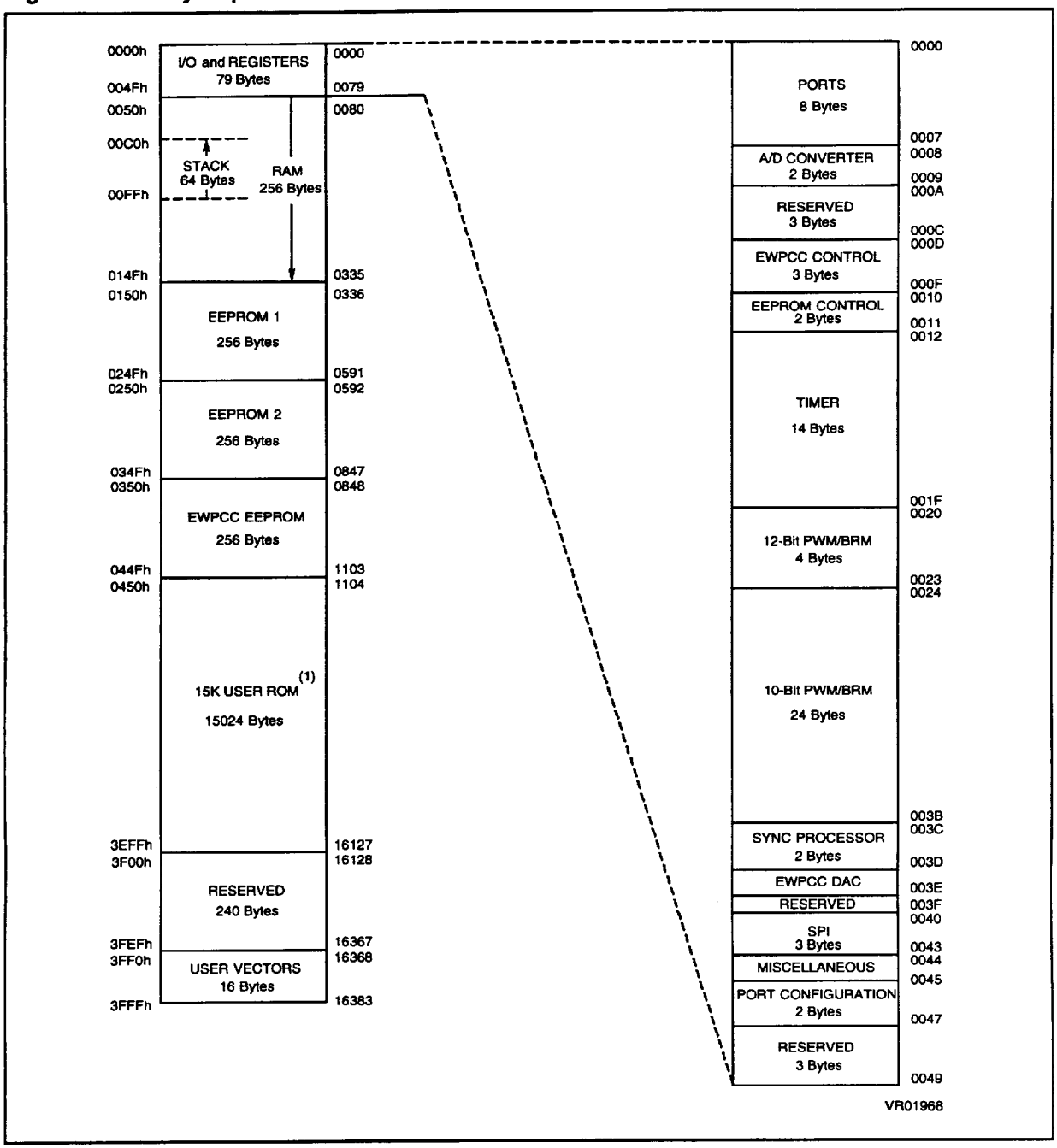
1.4 MEMORY MAP

As shown in Figure 5, the MCU is capable of addressing 16K bytes of memory and I/O registers. In the ST7271, 16383 of these bytes are user accessible.

The available memory locations consist of 80 bytes of I/O registers, 256 bytes of RAM, 512 bytes of EEPROM, 256 bytes of EWPCC EEPROM and 15Kbytes of user ROM. The RAM space includes 64 bytes for the stack from 00FFh to 00C0h.

The highest address bytes contain the user defined reset and interrupt vectors

Figure 5. Memory Map



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

1.5 I/O and REGISTER MAP

Address	Register Name
0000h	PORT A DATA REGISTER
0001h	PORT B DATA REGISTER
0002h	PORT C DATA REGISTER
0003h	PORT D DATA REGISTER
0004h	PORT A DATA DIRECTION REGISTER
0005h	PORT B DATA DIRECTION REGISTER
0006h	PORT C DATA DIRECTION REGISTER
0007h	PORT D DATA DIRECTION REGISTER
0008h	A/D DATA REGISTER
0009h	A/D CONTROL/STATUS REGISTER
000Ah-000Ch	Reserved
000Dh	EWPCC0 REGISTER
000Eh	EWPCC1 REGISTER
000Fh	EWPCC EEPROM CONTROL REGISTER
0010h	EEPROM 1 CONTROL REGISTER
0011h	EEPROM 2 CONTROL REGISTER
0012h	TIMER CONTROL REGISTER
0013h	TIMER STATUS REGISTER
0014h	INPUT CAPTURE REGISTER 1, High
0015h	INPUT CAPTURE REGISTER 1, Low
0016h	OUTPUT COMPARE REGISTER 1, High
0017h	OUTPUT COMPARE REGISTER 1, Low
0018h	COUNTER REGISTER, High
0019h	COUNTER REGISTER, Low
001Ah	ALTERNATE COUNTER REGISTER, High
001Bh	ALTERNATE COUNTER REGISTER, Low
001Ch	INPUT CAPTURE REGISTER 2, High
001Dh	INPUT CAPTURE REGISTER 2, Low
001Eh	OUTPUT COMPARE REGISTER 2, High
001Fh	OUTPUT COMPARE REGISTER 2, Low
0020h	PWM0 - 12 bit PWM/BRM
0021h	BRM0 - 12 bit PWM/BRM
0022h	PWM1 - 12 bit PWM/BRM
0023h	BRM1 - 12 bit PWM/BRM
0024h	PWM2 - 10 bit PWM/BRM
0025h	BRM3+BRM2

Address	Register Name
0026h	PWM3
0027h	PWM4
0028h	BRM5+BRM4
0029h	PWM5
002Ah	PWM6
002Bh	BRM7+BRM6
002Ch	PWM7
002Dh	PWM8
002Eh	BRM9+BRM8
002Fh	PWM9
0030h	PWM10
0031h	BRM11+BRM10
0032h	PWM11
0033h	PWM12
0034h	BRM13+BRM12
0035h	PWM13
0036h	PWM14
0037h	BRM15+BRM14
0038h	PWM15
0039h	PWM16
003Ah	BRM17+BRM16
003Bh	PWM17
003Ch	SYNC MUX CONTROL REGISTER
003Dh	SYNC COUNTER CONTROL REGISTER
003Eh	EWPCC DAC REGISTER
003Fh	Reserved
0040h	SPI DATA I/O REGISTER
0041h	Reserved
0042h	SPI CONTROL REGISTER
0043h	SPI STATUS REGISTER
0044h	MISCELLANEOUS REGISTER
0045h	PORT B CONFIGURATION REGISTER
0046h	PROGRAMMABLE INPUT/OUTPUT CONFIGURATION REGISTER
0047h	Reserved
0048h	Reserved
0049h	Reserved

1.6 WATCHDOG SYSTEM

The watchdog system consists in a divider-by-8 counter and a fixed divide-by-1024 prescaler. It is controlled through bit WDOG of the Miscellaneous Register.

Two mask options are provided:

- The WATCHDOG ENABLE mode mask option
- The WATCHDOG DURING WAIT mask option

The Watchdog Enable Mode mask option selects the state of the watchdog system after an external or a power-on reset. In the "programmable enable" option, a reset causes the watchdog to be disabled and the counter to be forced to zero. In the "auto enable" option, the watchdog is automatically enabled after the start-up procedure.

When the watchdog is configured with the "programmable enable" option, the watchdog system is enabled by setting the WDOG bit of the Miscellaneous Register (0044h). Only an external or a power-on reset can clear WDOG and disable the watchdog system.

Whatever the option, when the watchdog counter is enabled, it is driven by the CPU clock through the divide-by-1024 prescaler (i.e. the counter clock period is 1024 CPU clock cycles). It is reset to zero by writing WDOG at 1. A system reset is generated if the counter reaches its maximum count (8). To avoid a system reset, the software must therefore regularly reset the counter at least before the watchdog time from the last clear or from the time the watchdog system has been enabled.

Care has to be taken when enabling the counter ("programmable enable" option only). The prescaler is in an unknown state at the time WDOG is set. The first rising edge can thus be sent to the watchdog counter after a time comprised between 0 and 1024 CPU clock cycles. In this mode, the first reset of the watchdog counter should therefore not occur later than 6x1024 to 7x1024 CPU clock cycles after it has been enabled.

The system reset is generated by pulling down the $\overline{\text{RESET}}$ pin for at least one and a half CPU clock cycle. The state of the $\overline{\text{RESET}}$ pin is re-entered to the reset logic, thus causing an external reset to be issued.

The Watchdog During Wait mask option determines the watchdog function during the WAIT low power mode. In the "active during WAIT" option, the watchdog is kept active, thus able to reset the MCU if it remains in WAIT mode longer than the watchdog timeout period. In the "suspended during WAIT" option, it suspends operation during the WAIT mode and resets its counter. It will then resume operation when exiting the WAIT mode.

The STOP mode is inhibited when the watchdog system is enabled. However if a STOP instruction is executed while it is enabled, a watchdog reset is immediately generated.

MISCELLANEOUS REGISTER (0044h)

Read/Write

Reset Value: 1111 1010 (FAh)

7							0
1	1	1	1	1	VSYNCI	INT	WDOG

b7-b3 = **Unused**, read "1" when accessed.

b2 = **VSYNCI:** Internal Vsync (Read-Only) This bit shows the state of the Vsync output by the Sync Processor.

b1 = **INT:** Interrupt Request. This bit sets the interrupt configuration for the PORT B wakeup Interrupt Request:

INT = 0 : selects the falling edge option only

INT = 1 : selects the falling edge or low-level option.

WARNING. This bit can only be written ONCE after reset. Writing to INT is disabled after the first write to the Miscellaneous Register. Bit manipulation instructions should be used with extreme caution when writing to this register.

b0 = **WDOG:** Watchdog System. Whatever the WATCHDOG ENABLE MODE mask option, the watchdog counter is reset when WDOG is written at 1. When the MCU is configured with the "programmable enable" option, the WDOG bit is low after a reset. It must be set to enable the watchdog system. Writing a '0' clears the WDOG bit, but does not change the watchdog condition. Only a reset can clear WDOG.

1.7 SYSTEM CLOCK

General description.

The MCU accepts either a Crystal/Ceramic resonator or an external clock to provide the internal oscillator.
The internal clock (f_{INT}) is derived by a divide-by-2 from the internal oscillator frequency (f_{OSC}).

Crystal. The internal oscillator is designed to interface with an AT cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC}. The circuit shown on Figure 6 is recommended when using a crystal. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.

Ceramic Resonator. A ceramic resonator may be used in place of the crystal in low cost applications. The circuit on Figure 6 is recommended when using a ceramic resonator. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

External Clock. An external clock should be applied to the OSCIN input with the OSCOUT pin not connected as shown figure 7. The Crystal clock specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used.

Figure 6. Crystal/Ceramic Resonator

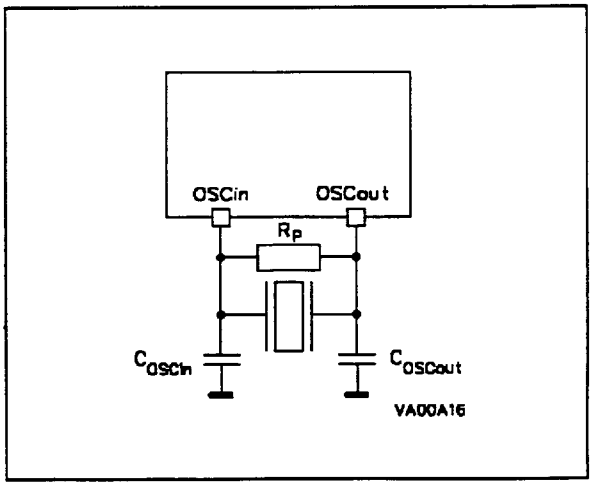


Figure 7. External Clock Source Connections

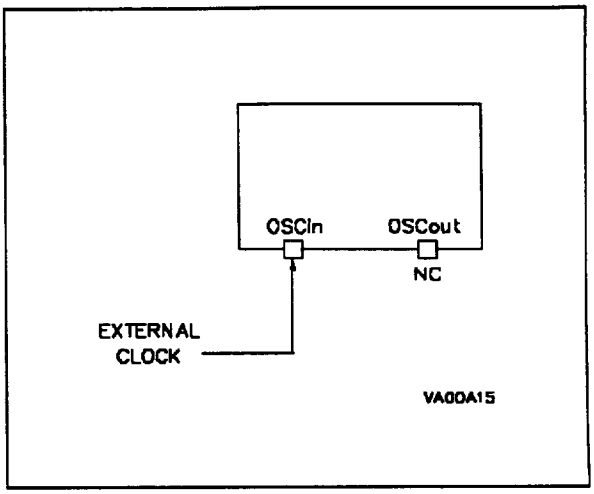
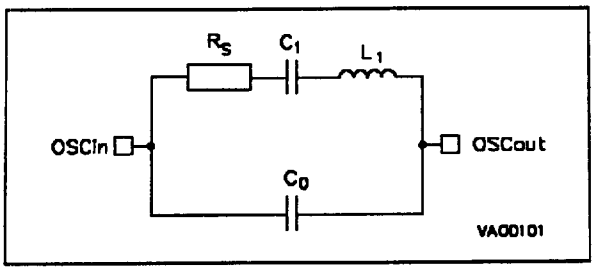


Figure 8. Equivalent Crystal Circuit



Recommended Settings for Crystal

	2MHz	4MHz	Unit
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	pF
C_{OSCin}	15-40	15-30	pF
C_{OSCout}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	10^3

Recommended Settings for Ceramic Resonator

	2-4MHz	Unit
R_{SMAX}	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSCin}	30	pF
C_{OSCout}	30	pF
R_P	1-10	MΩ
Q	1250	

1.8 RESET AND INTERRUPTS

The Reset procedure is used to provide an orderly software startup or to quit low power modes.

Two reset modes are provided: a power-on reset and an external reset at the $\overline{RESET}$ pin. The watchdog reset is considered as an external interrupt as the watchdog system generates the MCU reset by pulling down the $\overline{RESET}$ pin.

1.8.1 Power-on Reset (POR)

The power-on reset is generated upon detection of a positive transition on V_{DD} (refer to Figure 9). It causes the reset vector to be fetched from addresses 3FFEh and 3FFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 4096 CPU clock cycle delay from the time that the oscillator becomes active. At the end of the power-on reset, the MCU can be maintained in the reset condition by holding the external reset low. The $\overline{RESET}$ pin can therefore be used to ensure V_{DD} has risen to a point where the MCU can properly operate before running the MCU program.

During the POR, the $\overline{RESET}$ pin is pulled low, thus permitting the MCU to reset other devices.

The power-on reset is strictly used for power up conditions and should not be used to detect any drop in the power supply voltage. There is no internal provision for a power-down reset.

1.8.2 External Reset

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

As shown in Figure 9, the $\overline{RESET}$ signal must stay low for a minimum of one and a half CPU clock cycles. A reset causes the reset vector to be fetched at addresses 3FFEh and 3FFFh in order to be loaded into the PC and with program execution starting from this point.

The external reset is also used by the watchdog system to reset the MCU. When active, the power-on reset circuitry pulls down the $\overline{RESET}$ pin. In both cases, the $\overline{RESET}$ pin can be used as an output to reset other devices. However, the pull-down circuitry includes current limitation to allow the connection of any input signal, including from an RC type circuit.

An internal Schmitt trigger at the $\overline{RESET}$ pin is provided to improve immunity to noise.

Table 2. List of sections affected by RESET,WAIT and STOP

Section	RESET	POR	WAIT	STOP	
Timer Prescaler reset to zero	X	X	-	-	
Timer Counter set to FFFCh	X	X	-	-	
All Timer enable bit set to 0 (disable)	X	X	-	-	
Data Direction Registers set to 0 (as Inputs)	X	X	-	-	
Set Stack Pointer to 00FFh	X	X	-	-	
Force Internal Address Bus to restart vector 3FFEh, 3FFFh	X	X	-	-	
Set Interrupt Mask Bit (I-Bit, CCR) to 1 (Interrupt Disable)	X	X	-	-	
Set Interrupt Mask Bit (I-Bit, CCR) to 0 (Interrupt Enable)	-	-	X	X	
Reset STOP Latch	X	-	-		
Reset INT Latch	X	-	-		
Reset WAIT Latch	X	-	-		
Disable Oscillator (for 4096 cycles)	-	X	-	X	
Set Timer Clock to 0	-	-	X		
Watchdog counter reset	X	-	X		
Watchdog WDOG-BIT reset	X	-	X		
EEPROM control bits reset	X	-	-		
PWM/BRM registers reset	X	-	-		
EWPCC DAC register reset	X	-	-		
SYNC registers reset	X	-	-		

1.8.3 Interrupts

The ST7271 may be interrupted by one of three different methods: three maskable hardware interrupts (PORT B, SPI or TIMER) and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 10.

The maskable interrupts must be enabled in order to be serviced. However, disabled interrupts can be latched and processed when they are enabled. When an interrupt has to be serviced, the PC, X, A and CC registers are saved onto the stack and the interrupt mask (I bit 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 of the interrupt to service and the interrupt service routine runs (refer to Table 3 for vector addresses). The interrupt service routine should finish with the IRET instruction which causes the contents of the registers to be recovered from the stack and normal processing to resume. Note that the I bit is then cleared if and only if the corresponding bit stored in the stack is zero.

Though many interrupts can be simultaneously pending, a priority order is defined (see Table 3). The $\overline{\text{RESET}}$ pin has the highest priority.

If the I bit is set, TRAP is the only enabled interrupt.

Interrupts allow the processor to leave the Wait low power mode.

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

PORTB Interrupt. The PORTB Interrupt can be generated on the falling edge of any pin of PB0-PB7 if it is defined as an interrupt source. When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then processed according to the flowchart on Figure 10.

If the interrupt is disabled (I high), the triggering edge of the wake-up interrupt sources logical-ORed is internally latched and the interrupt

Figure 9. Reset Timing Diagram

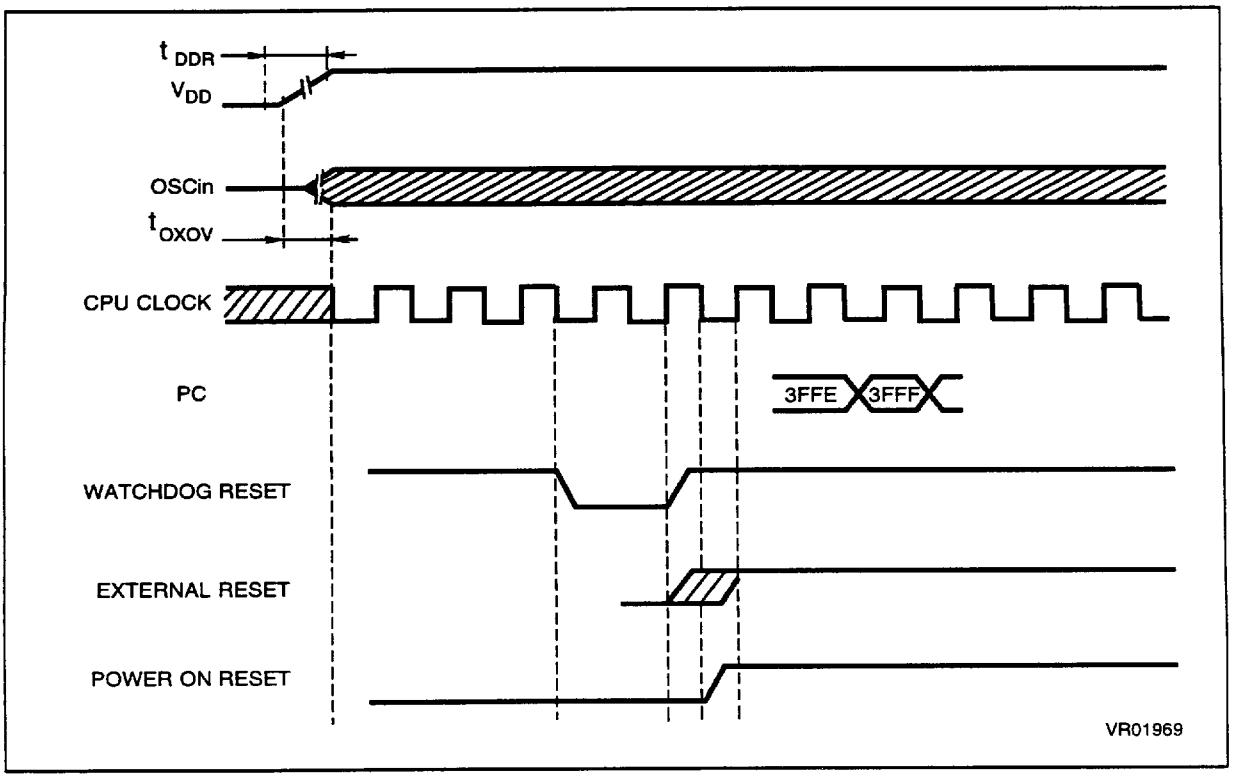
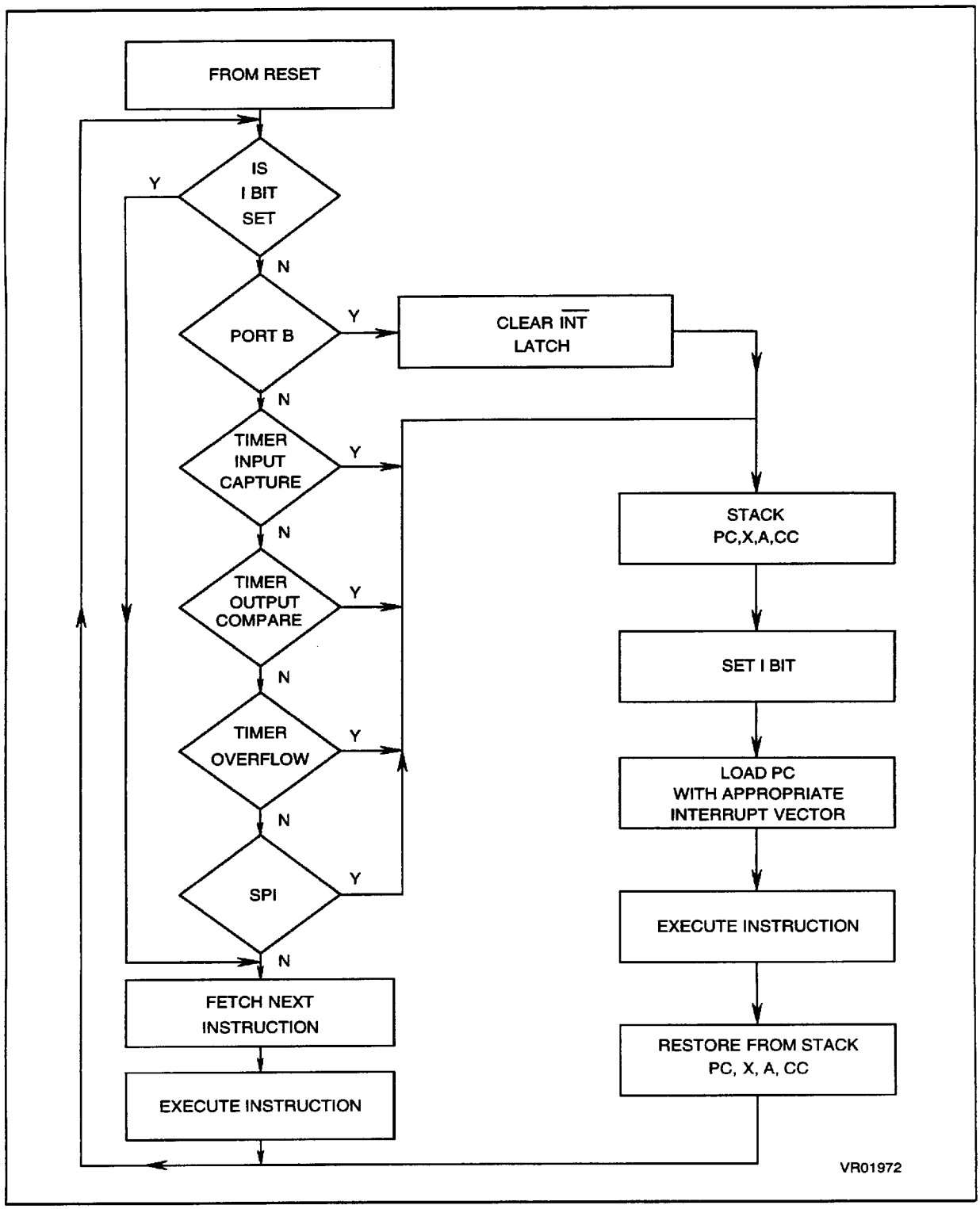


Figure 10. Interrupt Processing Flow-Chart



remains pending to be processed as soon as the interrupt is enabled. This internal latch is cleared in the first part of the service routine. Therefore, only one external interrupt can be latched and serviced as soon as possible. When the PORTB wake-up function is enabled, care should be taken after a Reset condition interrupt. The flowchart proposed in Figure 11 should be used in these cases.

Timer Interrupt. Five different timer interrupt flags are able to cause a timer interrupt when they are active if both the I bit of the CCR is reset and if the corresponding enable bit is set. If either of these conditions is false, the interrupt is latched and thus remains pending.

The interrupt flags are located in the Timer Status Register (0013h). The Enable bit are in the Timer Control Register (0012h).

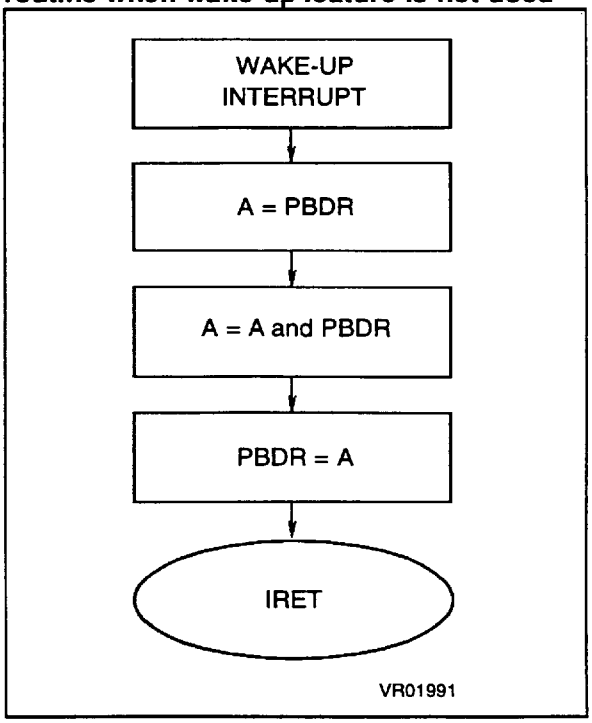
When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then serviced according to the flowchart on Figure 10. Software in the timer service routine must determine the priority and cause of the timer interrupt by examining the interrupt flags and the status bits located in the TSR.

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

Serial Peripheral Interface (SPI) Interrupts

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location 0043h) is set, provided the I bit in the condition code register

Figure 11. Proposed Wake-up Interrupt routine when wake-up feature is not used



cleared and the enable bit in the serial peripheral control register (location 0042h) is enabled.

When the interrrupt is recognized, the current state of the machine is pushed into the stack and I bit in the condition code register is set. These masks further interrupt until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location 3FF2 and 3FF3 which contains the starting address of the interrupt service routine.

Table 3. Interrupt and Reset priorities

Vector Address	Interrupt Source	Masked by	Priority	
3FFEh,3FFFh	RESET and POWER-ON (POR)		none	Highest
3FFCh,3FFDh	SOFTWARE Interrupt (TRAP)		none	↑
3FFAh,3FFBh	PORT B Wake up		I-Bit	
3FF8h,3FF9h	TIMER INPUT Capture (1 and 2)		I-Bit	
3FF6h,3FF7h	TIMER OUTPUT Compares (1 and 2)		I-Bit	
3FF4h,3FF5h	TIMER OVERFLOW		I-Bit	
3FF2h,3FF3h	SPI		I-Bit	
3FF0h,3FF1h	Reserved		-	Lowest

1.9 LOW POWER MODES

STOP Mode. The STOP mode is the MCU lowest power consumption mode. The STOP mode is entered by executing the STOP instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals. The STOP mode cannot be used when the watchdog is enabled, if the STOP instruction is executed while the watchdog system is enabled, a watchdog reset is generated thus resetting the entire MCU.

When entering the STOP mode, the I bit in the Condition Code Register cleared. Thus, the external interrupts are allowed and the MCU is placed at its nominal speed (see CLOCK SYSTEM). All other registers and memory remain unaltered and all I/O lines remain unchanged.

The MCU can exit the STOP mode upon reception of either an external interrupt on PORTB or a power-on or external reset. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

WAIT Mode. This mode is a low power consumption mode, but the power consumption is higher than in the STOP mode. The consumption can be further reduced by entering the slow mode.

The WFI instruction places the MCU in the WAIT mode.

In the WAIT mode, the internal clock remains active but all CPU processing is stopped; however, all other peripherals are still running. The watchdog can either be active or not according to the WATCHDOG DURING WAIT mask option.

During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts.

All other registers and memory remain unaltered and all parallel I/O lines remain unchanged.

An interrupt or a reset causes the MCU to exit the WAIT mode. An interrupt while the MCU is in the WAIT mode causes the corresponding interrupt vector to be fetched, the interrupt routine to be executed and normal processing to resume. A reset causes the program counter to fetch the reset vector and processing starts as for a normal reset.

Table 2 gives a list of the different sections affected by the low power modes. For detailed information on a particular devices, please refer to the corresponding parts.

Figure 12a. STOP Flow Chart

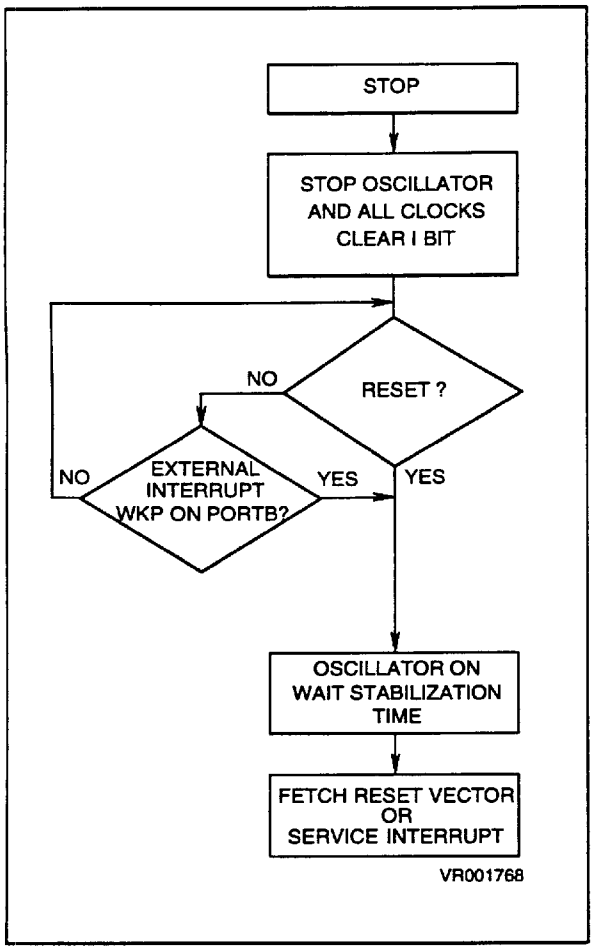
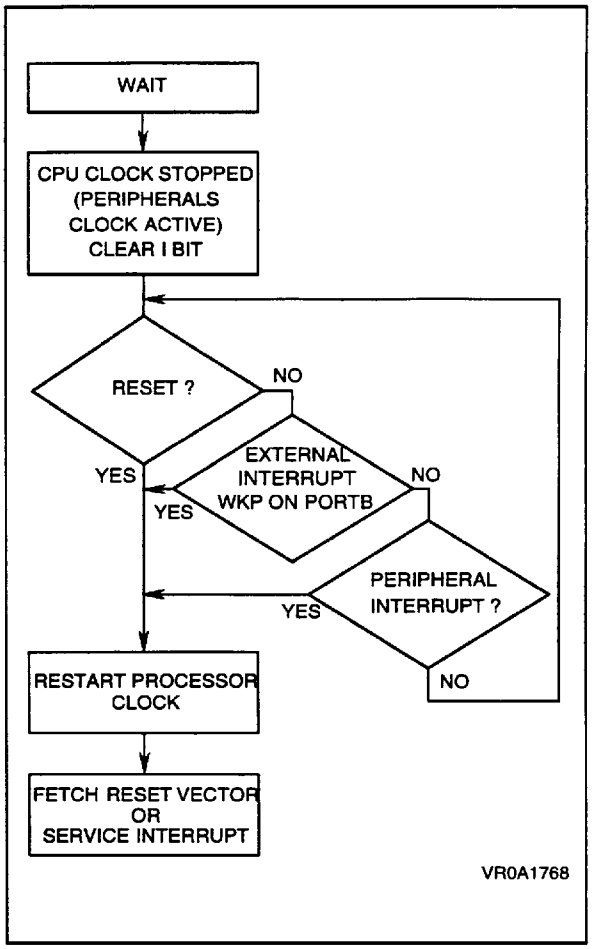


Figure 12b. WAIT Flow Chart



NOTES

PAGE(S) INTENTIONALLY BLANK

2.1 EEPROM

2.1.1 Introduction

The on-chip EEPROM provides a non-volatile storage of user programmed data. It is read as a normal Read-only memory location, however user programs must not run from the EEPROM.

Programming and erasure are made in conjuction with the EEPROM control registers while 8 data latches allow simultaneous write or erase of 1 to 8 bytes in the EEPROM memory in the same programming cycle. The constraint is that all addressed memory bytes must be on the same row of the EEPROM memory array, that is up to eight bytes with the address bits A7, A6, A5, A4 and A3 constant, and with A2, A1 and A0 selecting the address(es) to be written within the row.

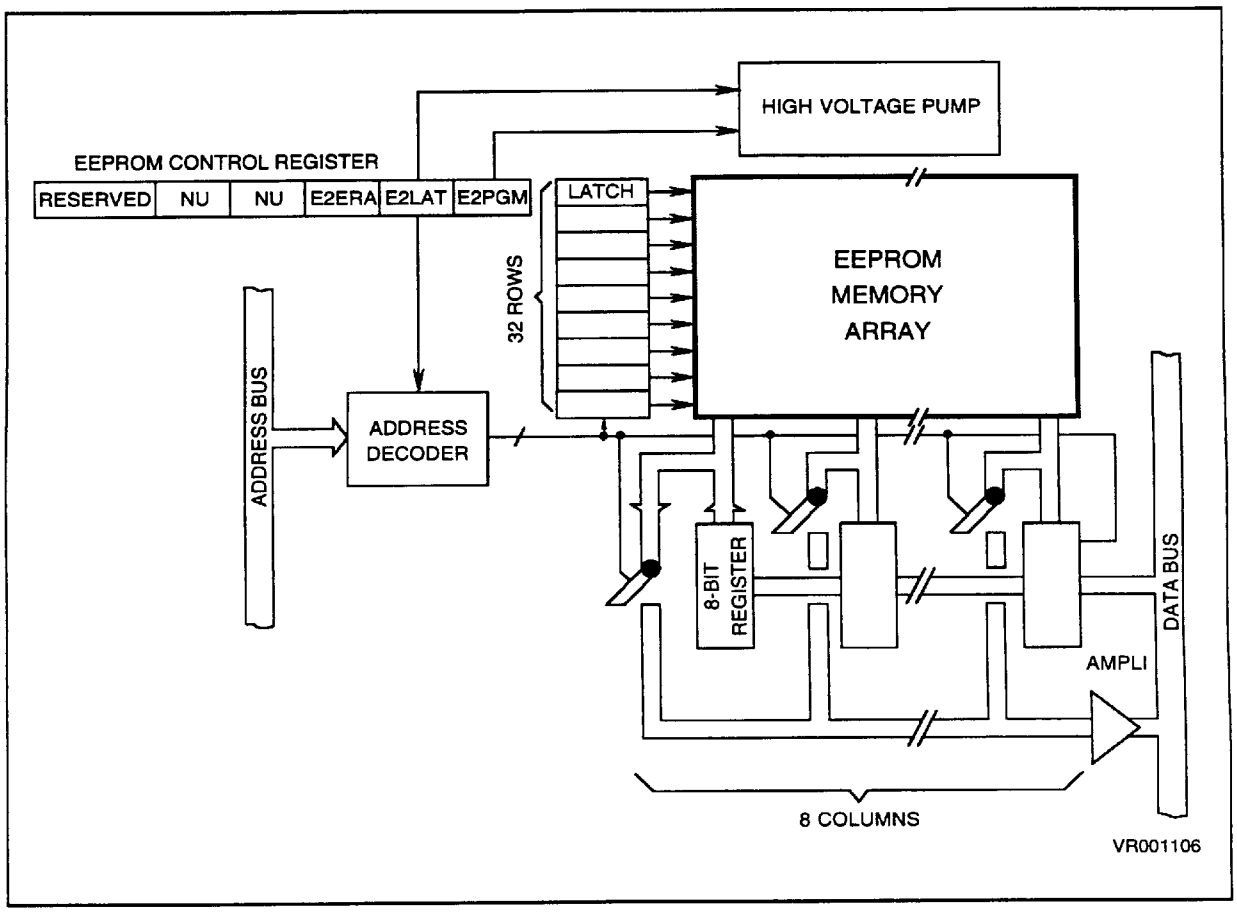
The EEPROM cell includes an internal charge pump to avoid the need of an external high voltage supply for the erase and programming functions.

2.1.2 Functional Description

As shown in Figure 13, the EEPROM is a 8 columns by 32 rows array. The row is selected by the A7, A6, A5, A4, A3 bits. Each column is associated to an 8-bit data register.

Read Operation (E2LAT="0").
The EEPROM can be read as a normal ROM location when the E2LAT bit of the Control Register is low. When E2LAT is low, the E2PGM and E2ERA bits are forced low.

Figure 12. EEPROM Block Diagram



EEPROM (Continued)

Write/Erase Operation. (E2LAT="1")

When E2LAT is set to "1", a write to an EEPROM location latches the data written in the 8-bit register corresponding to the decoded column and sets an internal flag for the decoded row.
As there are 8 columns in each row, up to 8 locations (having the same A7, A6, A5, A4, A3 address bits) can be simultaneously written or erased.

To **erase** bytes, the E2LAT and E2ERA bits are set to "1", and an instruction is made to write to the EEPROM addresses to be erased (the data value is not significant). The E2PGM bit must have been set after this operation to turn the charge pump on.

To **write** bytes, the E2LAT bit is set to "1", and the data is written to the appropriate EEPROM address(es). The E2PGM bit must have been set after this operation to turn the charge pump on.

WARNING a minimum delay of 20µs must be maintained after a programming operation (the falling edge of E2LAT) before the next read or write of the EEPROM. This time is required to discharge the high voltage in the array.

Notes

- Each block of 256 bytes of EEPROM is controlled by an independant EEPROM Control Register. Please refer to the Memory Map for both EEPROM memory block locations.
- It is mandatory to erase bytes before writing them.
- E2LAT must be kept high for the programming time tprog and then be cleared.
- When E2LAT is high, access to the EEPROM array is not possible.
- It is not allowed to perform successive write or erase cycles without clearing E2LAT between each write/erase (see Warning above).

2.1.3 Register Description

EEPROM CONTROL REGISTER 1 (0010h)

EEPROM CONTROL REGISTER 2 (0011h)

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

7							0
Res	Res	Res	—	—	E2ERA	E2LAT	E2PGM

This register contains the bits required to read, erase and program the EEPROM. They are defined as follow:

b7-5 = **Reserved**, must be held to "0"

b4,3 = **Unused**

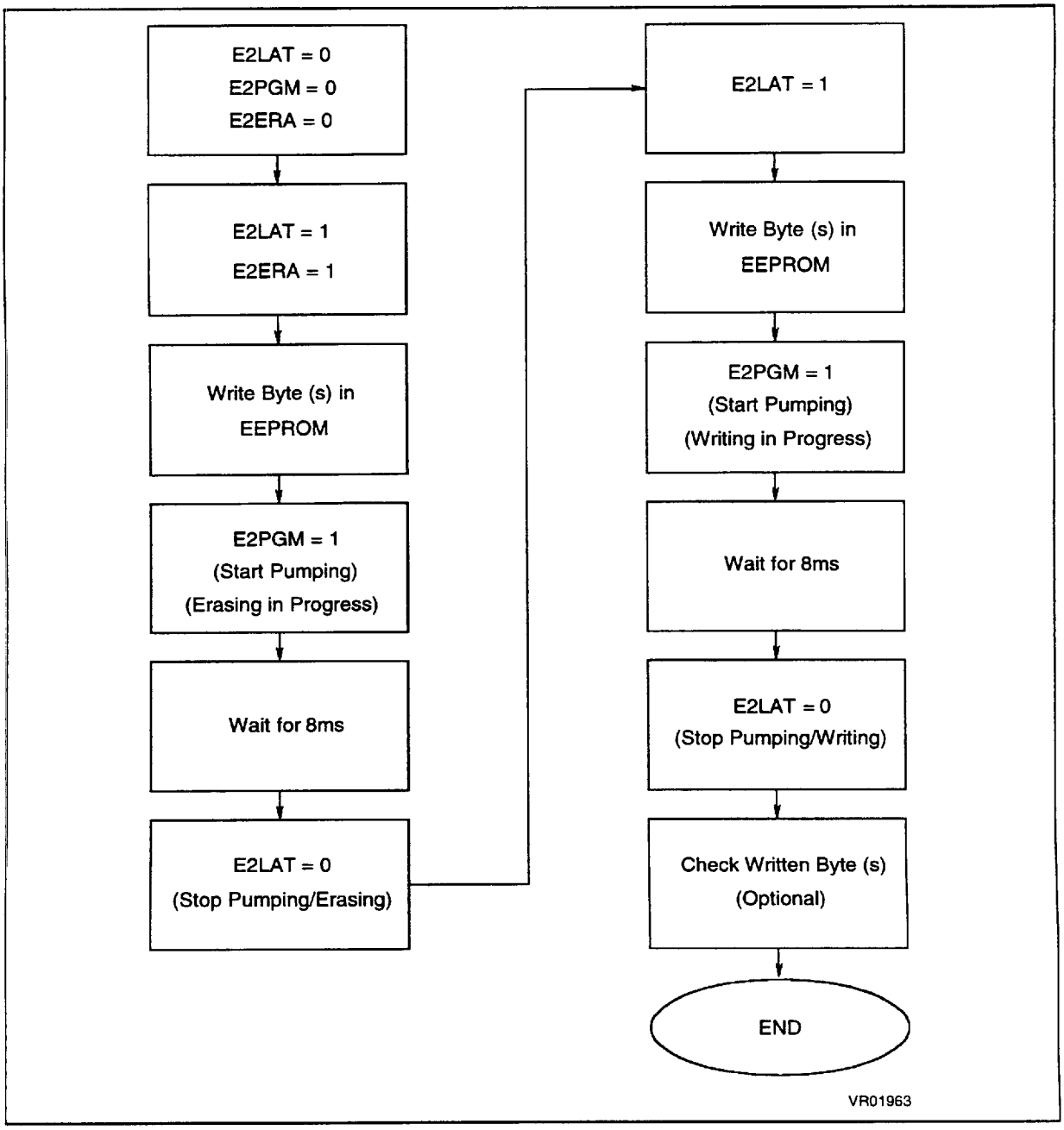
b2 = **E2ERA:** EEPROM Erase. E2ERA must be set to "1" for an erase operation. It must be set after or at the same time as E2LAT. It cannot be changed once an EEPROM address is selected. It is held low when E2LAT is low. It is therefore automatically reset when E2LAT is reset.

b1 = **E2LAT:** EEPROM Latch Enable. When E2LAT is reset to "0", data can be read from the EEPROM. When it is set to "1" and E2PGM reset to "0", a write into the EEPROM array causes the data to be latched, according to the address into one of 8 data registers. An additional internal flag is latched to select the row. The selected columns and row determine the locations involved in the next erase or programming operation. E2LAT must be cleared after each programming or erase operation. E2ERA and E2PGM are forced low when E2LAT is low.

b0 = **E2PGM:** EEPROM Program Mode. This bit allows the internal charge pump to be switched on or off. When set to "1", the charge pump generator is on and the high voltage is applied to the EEPROM array. When low, the charge pump generator is off. E2PGM can only be reset by resetting E2LAT.

EEPROM (Continued)

Figure 13. Programming Flow-chart of a basic routine



2.2 I/O PORTS

2.2.1 Introduction

The I/O ports allow the transfer of data through digital inputs and outputs, and, for specific pins, the input of analog signals or the Input/Output of dedicated signals for the on-chip peripherals (e.g. SPI, EWPCC and Timer). Please refer to the following table for a summary of these Alternate functions.

2.2.2 Functional Description

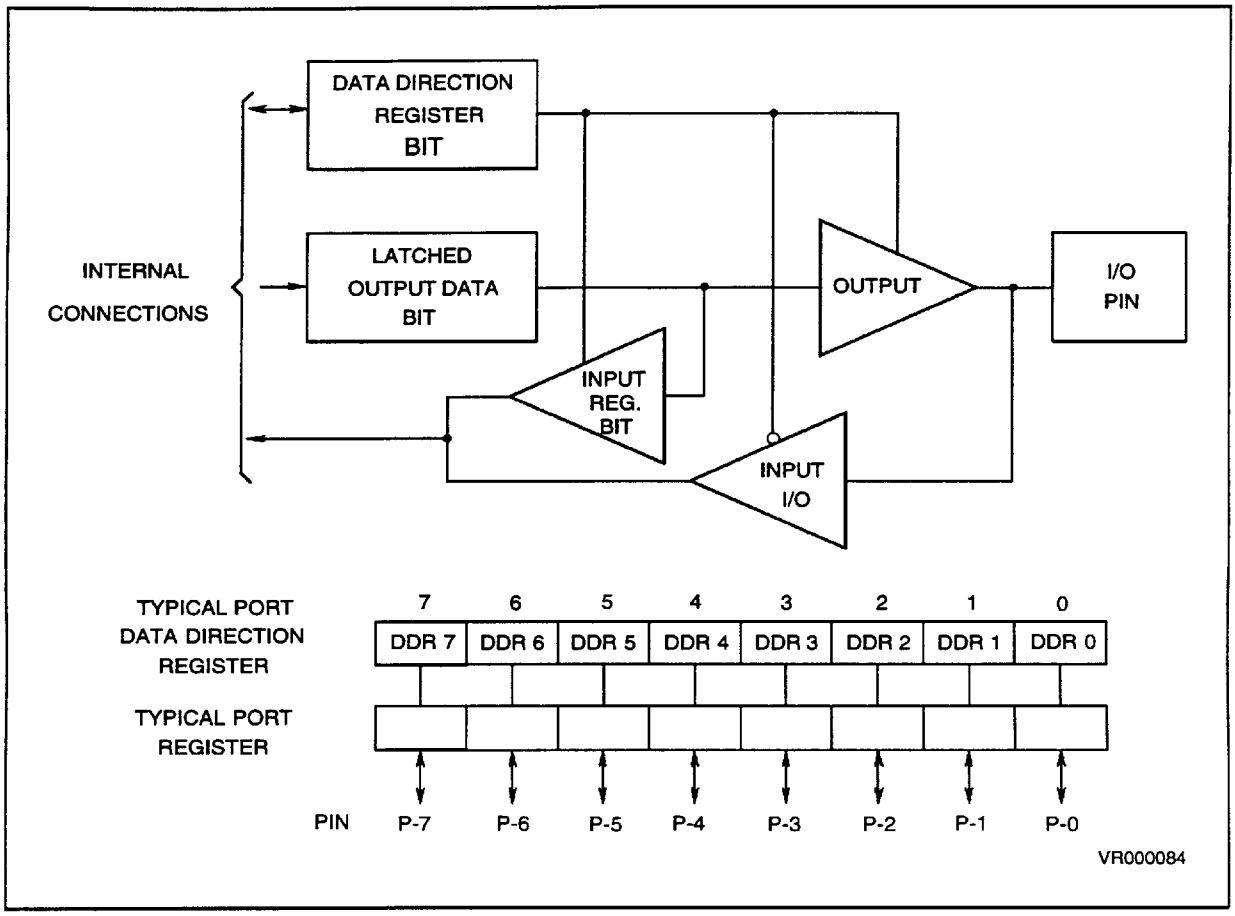
Each port pin of the I/O Ports can be individually configured under software control as either input or output. Ports A, B are 8-bit I/O ports, Port C is a 6-bit I/O port and Port D is a 5-bit port.

Each bit of a 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 output and must be cleared to configure its associated pin as input. The Data Direction Registers can be written or read.

The typical I/O circuit is shown on Figure 15. Any write to an I/O port updates the port output register even if it is configured as input. Any read of an I/O port returns either the data latched in the port output register (pins configured as output) or the value at the I/O pin (pins configured as input) (see Table 6).

At power-on or external reset, all DDR registers are cleared, which configures all Ports B, C and D pins as inputs with pull-ups and Port A as inputs without pull-ups but the port output registers are not initialized. Thus, the I/O port should be written before setting the DDR bit to avoid undefined output levels.

Figure 14. I/O Pin Typical Circuit



I/O PORTS (Continued)

PORT A

Each Port A bit can be defined as an Input line (no pull-up) or as an Output Open drain line capable of handling typical current Isink of 10 mA for LED driving.

PORT B

Each bit of PORT B bit can be used as the Analog source to the Analog to Digital converter by selecting each individual bit independently in the Port B Configuration Register (address 0045h).
When the Analog function is selected for an I/O pin, the pull-up of the respective pin of Port B is disconnected and both the Data and Direction (DDR) registers of the respective pin are reset. Any further accesses to the respective DDR bit is blocked until the pin status is returned to normal I/O.

PORT B bit can also be configured on a bit basis as a wake-up interrupt input with an internal pull-up resistor. This mode is enabled by setting the corresponding Port B bit as a DIGITAL input (its bit in DDR set to '0' and its Analog function disabled) and the corresponding bit in the Port B Data Register must be set to '1'.
When this bit is subsequently forced low, an interrupt will be generated according to the status of the INT bit in the Miscellaneous Register.

Port B, bit 0 is only available for output if the East-West Pin-Cushion controller (EWPCC) is not used. If the EWPCC function is selected, Port B bit 0 MUST be set as input to enable the VFBACK timing input.

The pins not available for the 42 pin package (but present for the 56 pin package) are internally connected as standard digital inputs with pull-ups enabled.

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

PORT C

The available port pins of Port C may be used as general purpose I/O or as the I/O pins of the on-chip SPI and Timer Output Compare.
When used as digital Input, pull-up resistors may be switched on for ALL Port C inputs by setting the PUPC bit of the Programmable Input/Output Configuration Register (PCR).
When used as output the Open Drain mode is automatically set if the SPI is disabled.

Port C, bit 0 is switched from the normal I/O functionality to the output of the Timer Output Compare signal by resetting to '0' the OCOP bit of the PCR.

When the SPI is enabled, Port C bits 2-5 output bits are forced to the push-pull output configuration for high speed data transmission, while pins set to input have the pull-up resistor disconnected, regardless of the state of PUPC.
The default condition of open drain output (SPI not enabled) allows software emulation of communication using the I^2C-bus protocol.

PORT D

The I/O pins of Port D normally are used for the input and output of video synchronization signals to the Sync Processor, but are set to I/O Input with pull-up upon reset. The I/O mode can be set individually for each port bit to Input with pull-up and output push-pull through the Port D DDR.

The configuration to support the Sync Processor required that the SYNOP bit of the PCR be reset to '0'; this enables Port D bits 0, 1 and 2 to the sync inputs and outputs.

Note that as these inputs are switched from normal I/O functionality, the video synchronization signals may also be monitored directly through the Port D Data Register for such tasks as checking for the presence of video signals or checking the polarity of Horizontal and Vertical synchronization signals (when the Sync Inputs are switched directly to the outputs using the multiplexors of the Sync Processor).

Table 6. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output the I/O pin.
1	0	The state of the I/O pin to read.
1	1	The I/O pin is in an output mode. The output data latch is read.

* R/$\overline{W}$ is an internal signal.

2.2.3 Register Description

DATA REGISTERS

Port A: 0000h
Port B: 0001h
Port C: 0002h
Port D: 0003h

Read/Write

Reset Value: Undefined

7			0				
MSB			LSB				

DATA DIRECTION REGISTERS

Port A: 0004h
Port B: 0005h
Port C: 0006h
Port D: 0007h

Read/Write

Reset Value: 0000 0000 (00h) (as inputs)

7			0				
MSB			LSB				

PORT B CONFIGURATION REGISTER (0045h)

Read/Write

Reset Value: 0000 0000 (00h)

7			0				
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

b7-0 = **AD7-AD0:** Port B Digital/Analog Input Configuration Bits. When AD#i is set (i = 7-0), the pull-up on the respective pin #i of Port B is disconnected and the pin is configured as analog input; otherwise the pull-up is connected and pin configured as digital input (RESET condition) with no power consumption in the A/D channel.

Note. On the 42-pin package option, PB7-PB4 are not externally connected and are internally configured as standard digital inputs with pull-up.

PROGRAMMABLE INPUT/OUTPUT CONFIGURATION REGISTER (0046h)

Programmable Input/Output Configuration

Read/Write

Reset Value: 1111 1000 (F8h)

7							0
Res	CLMOP	OCOP	SYNOP	PUPC	POC2	POC1	POC0

b7 = **Res:** Reserved.

b6 = **CLMOP:** Clamping Signal Output Select. This bit selects either the PD3 I/O Pin Option or the output of the Clamping signal.

CLMOP = 0 : Clamping Signal

CLMOP = 1 : PD3 as Pull-up Input or Push-pull Output

b5 = **OCOP:** Timer Output Compare Select. This bit selects either the PC0 I/O Pin Option or the output of the Timer Output Compare.

OCOP = 0: Timer Output Compare

OCOP = 1: PC0 Input (with/no pull-up) or Output (push-pull)

b4 = **SYNOP:** SYNC Processor Function Select. This bit selects either the Sync Processor or PD0, PD1, PD2 I/O Pin Options.

SYNOP = 0: PD0 = CSYNCI, PD1 = HSYNCO and PD2 = VSYNCO

SYNOP = 1: PD0/PD1/PD2 Inputs (Pull-up) OR Outputs (Push-Pull).

Note. HSYNCO and VSYNCO can be directly read as Port bits by configuring PD1 and PD2 as inputs.

b3 = **PUPC:** PORT C Input Configuration Bit. This bit selects the input configuration for present bits of I/O Port C.

PUPC = 1 : Pull-up

PUPC = 0 : No Pull-up.

Whenever the SPI is active, the pull-up is disconnected from SPI input pins regardless of the state of PUPC and SPI outputs are set to push-pull.

b2-b0 = **POC2-POC0:** PWM/BRM Output Configuration Bits. These bits select the PWM/BRM output configuration.

PWM Group Channels		Value	
		0	1
A2 DA1, D3-6	P0C0	push-pull	open drain
B1 DA7-11	P0C1	push-pull	open drain
B2 DA12-DA17	P0C2	push-pull	open drain

In the case of uncomplete ports (Port C and Port D), non-implemented bits are read '0' whenever accessed.

2.3 16 BIT TIMER

2.3.1 Introduction

The 16-bit programmable timer consists of a 16-bit free running counter driven by a prescaler and control logic for two input captures and two output compare registers. It can be used for many purposes including pulse length measurement of input signals and generation of one output waveform.

The two input capture functions are dedicated to the timing functions of the Sync Processor and are internally connected to this source. They are thus not available for timing of other external signals.

When used with an 8MHz external oscillator frequency, the timer has a resolution of 0.5µs.

2.3.2 Functional Description

As the timer has a 16-bit architecture, each of its specific function blocks is represented by two registers. These registers contain the high order byte and low order byte of that function. However any access to the high order byte inhibits that specific timer capability until the low order byte is also accessed.

Note that correct software procedures should set the I bit of the Condition Code Register before accessing the high order byte to prevent an interrupt from occurring between the accesses to the high and low order bytes of any register.

Counter. The key element of the programmable timer is a 16-bit free running counter or counter register. It is preceded by a prescaler which divides the internal clock by two giving an operational frequency of 2MHz.

Software can read the counter at any time without affecting its value. It can be read from two locations, the Counter Register (0018h, 0019h) and Alternate Counter Register (001Ah, 001Bh). The only difference between these two read-only registers is the way the overflow flag TOF is handled during a read sequence.

A read sequence containing only a read of the least significant byte of the free running counter (from either the Counter Register or the Alternate Counter Register) will receive the LSB of the count value at the time of the read. A read of the most significant byte (from either the Counter Register or the Alternate Counter Register) simultaneously returns the MSB of the count value and causes the LSB to be transferred into a buffer.

The buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times. The read sequence is completed by reading the free running counter LSB, which actually returns the buffered value.

As shown in Figure 17 the free running counter is configured to FFFCh during reset, after $\overline{\text{RESET}}$ goes high. During a power-on reset (POR), the counter is also configured to FFFCh and begins running after the oscillator startup delay.

When the counter rolls over from FFFFh to 0000h, the Timer Overflow flag (TOF) of the Timer Status Register (TSR) is set. A timer interrupt is then generated if the TOIE enable bit of the Timer Control Register (TCR) is set, provided the I bit of the CCR is cleared. If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true. The interrupt request is cleared by reading TSR while TOF is set followed by an access (read or write) to the LSB of the Counter Register.

The TOF flag is not affected by accesses to the Alternate Counter Register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure on elapsed time) without risking the clearing of the TOF flag erroneously. Accesses to the timer without the intention of servicing the TOF flag should therefore be performed to the Alternate Counter Register while only the TOF service routine accesses the Counter Register.

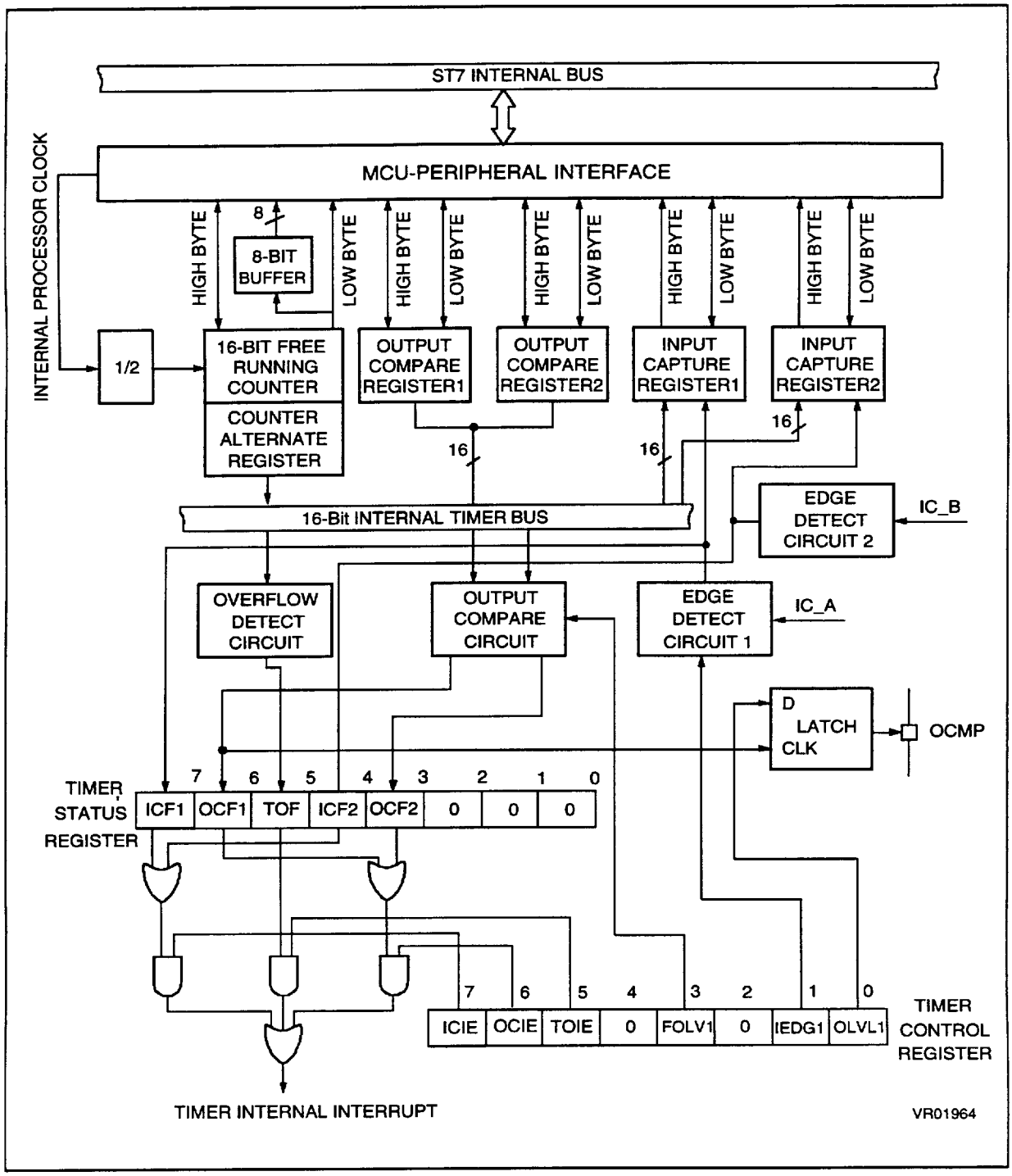
The free running counter can be reset under software control. This is performed by writing to the LSB of either the Counter Register or the Alternate Counter Register. The counter and the prescaler are then configured to their reset conditions. This reset also completes any 16-bit access sequence. All flags and enable bits are unchanged.

The value in the counter registers repeats every 131072 internal processor clock cycles (32 ms for f_{INT} = 4 MHz). As shown in Figure 17, the counter increment is triggered by a falling edge of the CPU clock.

The timer is not affected by the WAIT mode. In the HALT mode, the counter stops counting until the mode is exited. Counting then resumes from previous count (MCU woken by an interrupt) or from reset count (MCU woken by a reset).

Input Capture.

Figure 15. Timer Block Diagram



16 BIT TIMER (Continued)

The ST7271 features two input capture registers and two input capture interrupt enable bits. The input capture inputs IC_A and IC_B are connected through the VSYNCI and HSYNCI (OR CSYNCI) input pins respectively. When the SYNC processor is not being used these pins may be used for the external input captures of the timer. The input on HSYNCI may optionally be passed through a /256 prescaler before being passed to the IC_B input capture.

Input Capture Register 1 (ICR1) is a 16-bit register made up of two 8-bit registers: the most significant byte register (ICHR1), located at 0014h, and the least significant byte register (ICLR1) located at 0015h.

ICR1 is a read-only register used to latch the value of the free running counter after a defined transition is sensed by the input capture edge detector at IC_A. This transition is software programmable through the IEDG1 bit of the Timer Control Register (TCR). When IEDG1 is set, a rising edge triggers the capture; when IEDG1 is low, the capture is triggered by a falling edge. Care must be taken with the external circuitry to avoid unwanted interrupts when changing the interrupt edge.

When an input capture occurs, the flag ICF1 in Timer Status Register (TSR) is set. An interrupt is requested if the interrupt enable bit ICIE of TCR is set, provided the I bit of the CCR is cleared. Otherwise, the interrupt remains pending until both conditions become true. It is cleared by reading the Timer Status Register TSR followed by an access (read or write) to the LSB of ICR1.

The result stored in ICR1 is one more than the value of the free running counter on the rising edge of the internal processor clock preceding the active transition at pin IC_A (see Figure 17). This delay is required for internal synchronization. Therefore, the timing resolution of the input capture system is one count of the free running counter, i.e. 2 internal clock cycles.

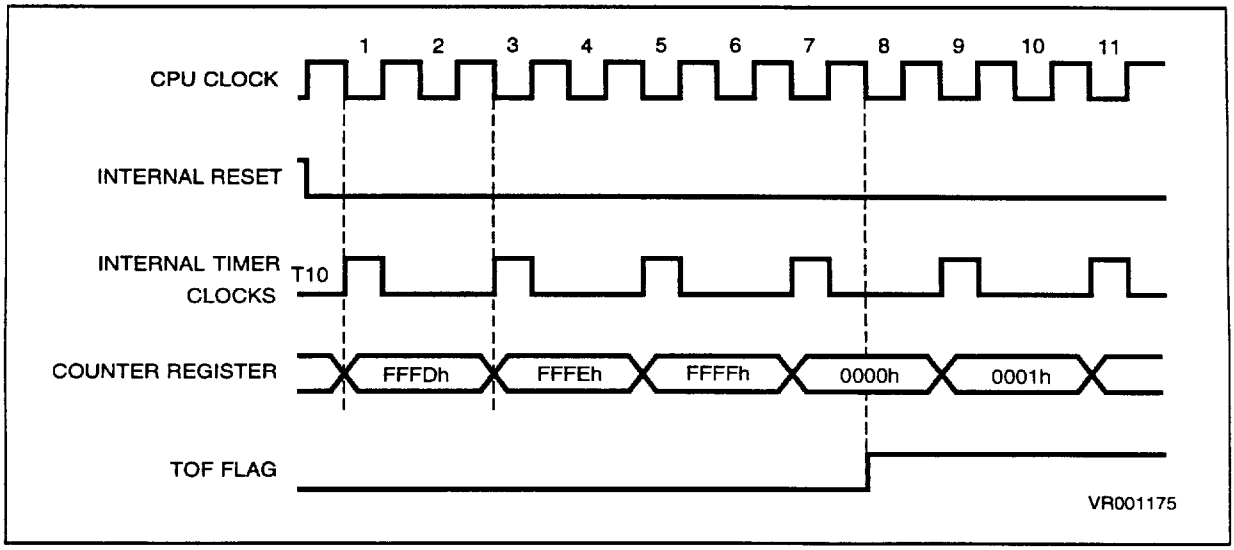
The free running counter is transferred to ICR1 on each proper signal transition regardless of whether the Input Capture Flag ICF1 is set or cleared. The ICR1 always contains the free running counter value which corresponds to the most recent input capture.

After a read of the MSB of ICR1 (ICHR1), counter transfer of input capture is inhibited until the LSB of ICR1 (ICLR1) is also read. This characteristic forces the minimum pulse period attainable to be determined by the time to service the interrupt and to execute the interrupt routine.

A read of ICLR1 does not inhibit the counter transfer. Again, minimum pulse periods are the ones which allow software to read the least significant byte and perform needed operations. There is no conflict between the read of ICR1 and the running counter transfer since they occur on opposite edges of the internal processor clock (see Figure 17).

The ICR1 is undetermined at power-on and is not affected by an external reset. Hardware circuitry has to provide protection from generating a wrong input capture when changing the edge sensitivity option of IC_A input through the IEDG1 bit.

Figure 16. Timer Timing Diagram



During the HALT mode, if at least one valid input capture edge occurs at the IC_A input, the input capture detect circuitry is armed. This action does not set any timer flags nor "wake-up" the MCU. If the MCU is awaken by an interrupt, there is an active input capture flag and data from the first valid edge that occurred during the HALT mode. If the HALT mode is exited by a reset, the input capture detect circuitry is reset and thus, any active edge that happened during the HALT mode is lost.

Input Capture Register 2 (ICR2) is a 16-bit register made up of two 8-bit registers: the most significant byte register (ICHR2), located at 001Ch, and the least significant byte register (ICLR2) located at 001Dh.

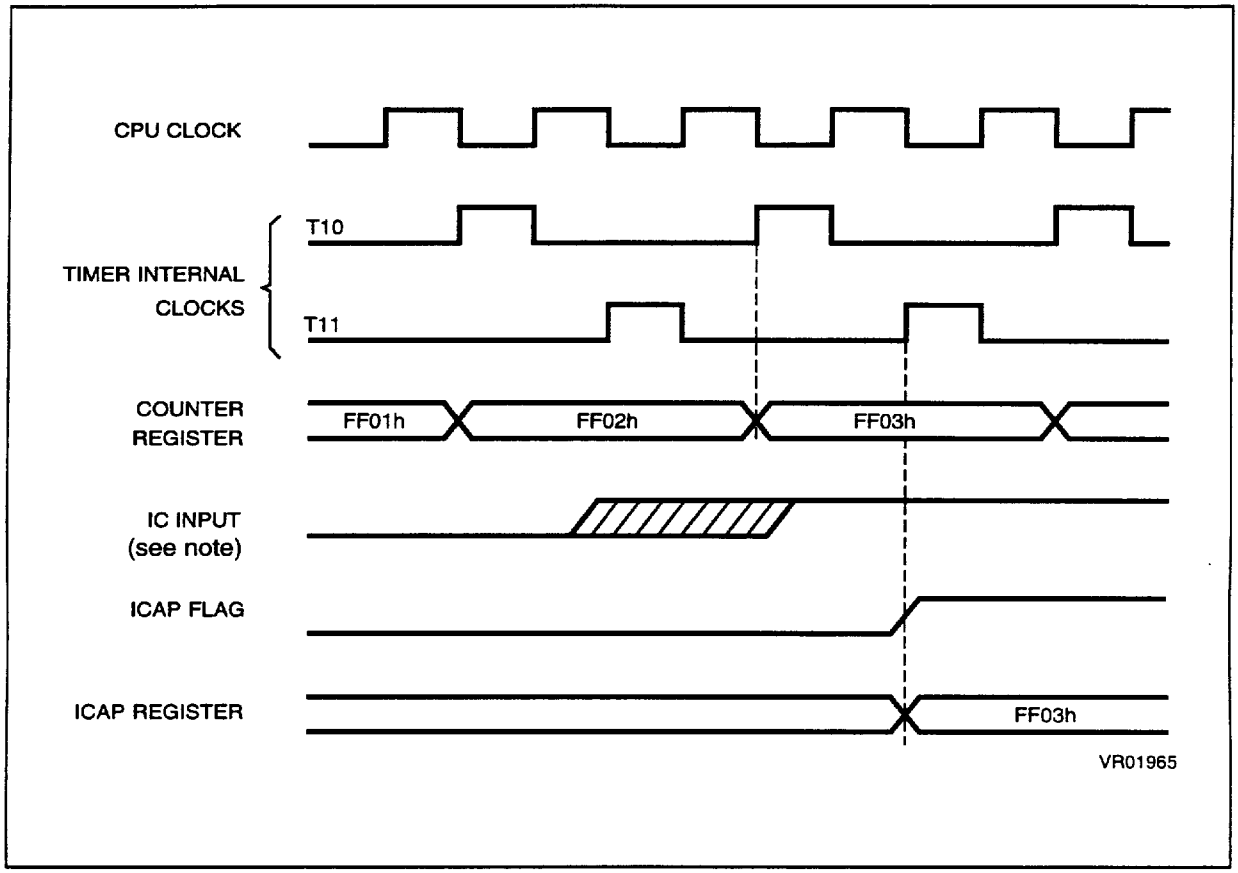
The previous description shown for Input Capture Register 1 is also applicable for the Input Capture Register 2, with the exception that Input Capture Register 2 is triggered only on a negative edge on input IC_B and with the substitution of the appropriate index in the bit and register names.

Output Compare.

There are two output compare registers: Output Compare Register 1 and 2 (OCR1 and OCR2). They can be used for several purposes such as controlling an output waveform or indicating when a period of time has elapsed. OCMP1 pin is associated with output compare 1; no pin is associated with Output Compare 2 which can be used for the generation of timer interrupts.

The Output Compare Registers are unique because all bits are readable and writable and are not affected by the timer hardware and reset. If a compare function is not used, the two bytes of the corresponding Output Compare Registers can be used as storage locations. Note that the same output compare interrupt enable bit is used for both output compares.

Figure 17. Input Capture Timing Diagram



Note. The diagram represents the case of a rising edge sensitivity (IEDG1=1). The capture is performed at the next rising edge of T11, if the action edge of ICAP happened before the previous T10 falling edge.

Output Compare Register 1. The Output Compare Register 1 (OCR1) is a 16-bit register, which is made up of two 8-bit registers: The most significant byte register (OCHR1) at address 0016h and the least significant byte register (OCLR1) at address 0017h.

The content of OCR1 is compared with the content of the free running counter once during every timer clock cycles, i.e. every 2 internal processor clock periods. If a match is found, the Output Compare Flag OCF1 of the TSR is set and the Output Level bit (OLVL1) of the TCR is clocked to the OCMP1 pin (see output compare timing diagram on Figure 19).

OLVL1 is copied to the corresponding output level latch and hence, to the OCMP1 pin regardless of whether the Output Compare Flag (OCF1) is set or not. The value in the OCR1 and the OLVL1 bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

An interrupt follows a successful output compare if the corresponding interrupt enable bit OCIE of the TCR is set, provided the I-bit of the CCR is cleared. Otherwise, the interrupt remains pending until both conditions are true. It is cleared by a read of TSR followed by an access to the LSB of the OCR1.

After a processor write cycle to the OCHR1 register, the output compare function is inhibited until the OCLR1 is also written. Thus, the user must write both bytes if the MSB is written first. A write made to only the LSB will not inhibit the compare function. The minimum time between two successive edges on the OCMP1 pin is a function of the software program.

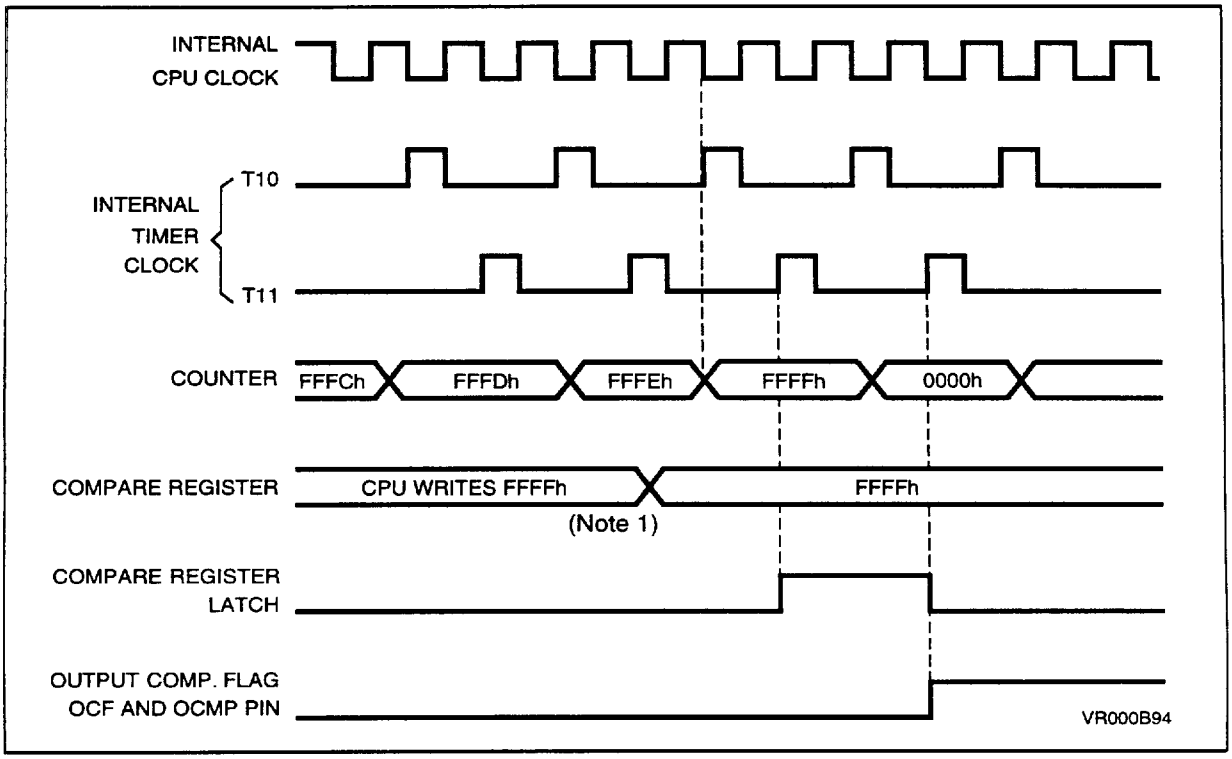
The OCMP1 output latch is forced low during reset and stays low until valid compares change it to a high level. Because the OCF1 flag and the OCR1 are undeterminate at power-on and are not affected by an external reset, care must be exercised when initiating the output compare function with software. The following procedure is recommended to prevent the OCF1 flag from being set between the time it is read and the write to OCR1:

Write to OCHR1 (further compares are inhibited).

Read the TSR (first step of the clearance of OCF1 [it may be already set]).

Write to OCLR1 (enables the output compare function and clears OCF1).

Figure 18. Output Compare Timing Diagram



Note 1. The CPU write to the compare registers may take place at any time but a compare only occurs at timer state T11. Thus a 2-cycles difference may exist between the write to the compare register and the actual compare.

Output Compare Register 2. The Output Compare Register 2 (OCR2) is a 16-bit register, which is made up of two 8-bit registers: the most significant byte register (OCHR2) at address 0001Eh and the least significant byte register (OCLR2) at address 0001Fh.

This register works as the Output Compare Register 1. For a complete description, please refer to the above and substitute the appropriate index in the bit and register names.

Software Force Compare. The force compare capability main purpose is to facilitate fixed frequency generation.

When the Force Output Level 1 bit (FOLV1) of TCR is written to 1, OLVL1 is copied to pin OCMP1. To provide this capability, internal logic allows a single instruction to change OLVL1 and causes a forced compare with the new value of OLVL1. OCF1 is not affected and thus, no interrupt request is generated.

2.3.3 Register Description

TIMER CONTROL REGISTER (0012h)

Read/Write
Reset Value: 0000 00x0 (00h or 02h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

b7 = **ICIE** Input Capture Interrupt Enable
If ICIE is set, a timer interrupt is enabled whenever the ICF1 status flags of TSR are set. If the ICIE bit is cleared, the interrupt is inhibited.

b6 = **OCIE:** Output Compare Interrupt Enable
If OCIE is set, a timer interrupt is enabled whenever the OCF1 or OCF2 status flags of TSR are set. If the OCIE bit is cleared, the interrupt is inhibited.

b5 = **TOIE:** Timer Overflow Interrupt Enable
If TOIE is set, a timer interrupt is enable whenever the TOF status flag of TSR is set. If the TOIE bit is cleared, the interrupt is inhibited.

b4 = **FOLV2:** Force Output Compare 2
This bit has no affect. FOLV2 is cleared by a system reset.

b3 = **FOLV1:** Force Output Compare 1
When written to 1, FOLV1 forces OLVL1 to be copied to the OCMP pin.

b2 = **OVL2**: Output Level 2. This bit has no affect.

b1 = **IEDG1:** Input Edge 1
The value of the IEDG1 determines which level transition on IC_A input will trigger a free running counter transfer to the ICR1. When IEDG1 is high, a rising edge triggers the capture since when low, a falling edge does.

b0 = **OLVL1** Output Level 1
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs at OCR1.

TIMER STATUS REGISTER (0013h)

Read Only
Reset Value: Undefined

7							0
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0

b7 = **ICF1** Input Capture Flag 1
ICF1 is set when a proper edge has been sensed by the input capture edge detector at IC_A. The edge is selected by the IEDG1-bit in TCR. ICF1 is cleared by a processor access to the TSR while ICF1 is set followed by an access (read or write) to the low byte of ICR1 (ICLR1).

b6 = **OCF1** Output Compare Flag 1
OCF1 is set when the content of the free running counter matches the content of OCR1. It is cleared by a processor access of TSR while OCF1 is set followed by an access (read or write) to the low byte of OCR1.

b5 = **TOF** Timer Overflow
TOF is set by a transition of the free running counter from FFFFh to 0000h. It is cleared by a processor access to TSR while TOF is set followed by an access (read or write) to the low byte of the counter low register. TOF is not affected by an access to the Alternate Counter Register.

b4 = **ICF2** Input Capture Flag 2
ICF2 is set when a negative edge has been sensed by the input capture edge detector at IC_B. ICF2 is cleared by a processor access to the TSR while ICF2 is set followed by an access (read or write) to the low byte of ICR2 (ICLR2).

b3 = **OCF2** Output Compare Flag 2
OCF2 is set when the content of the free running counter matches the content of OCR2. It is cleared by a processor access of TSR while OCF2 is set followed by an access (read or write) to the low byte of OCR2.

b2-0 = **Unused**, read as '0'.

16 BIT TIMER (Continued)

INPUT CAPTURE REGISTER 1, High Byte (0014h)

Read Only
Reset Value: Undefined

7							0
IC1.15	IC1.14	IC1.13	IC1.12	IC1.11	IC1.10	IC1.9	IC1.8

INPUT CAPTURE REGISTER 1, Low byte (0015h)

Read Only
Reset Value: Undefined

7							0
IC1.7	IC1.6	IC1.5	IC1.4	IC1.3	IC1.2	IC1.1	IC1.0

OUTPUT COMPARE REGISTER 1, High byte (0016h)

Read/Write
Reset Value: Undefined

7							0
OC1.15	OC1.14	OC1.13	OC1.12	OC1.11	OC1.10	OC1.9	OC1.8

OUTPUT COMPARE REGISTER 1, Low byte (0017h)

Read/Write
Reset Value: Undefined

7							0
OC1.7	OC1.6	OC1.5	OC1.4	OC1.3	OC1.2	OC1.1	OC1.0

COUNTER REGISTER, High byte (0018h)

Read Only
Reset Value: 1111 1111 (FFh)

7							0
C1.15	C.14	C.13	C.12	C.11	C.10	C.9	C.8

COUNTER REGISTER, Low byte (0019h)

Read/Write
Reset Value: 1111 1100 (FCh)

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

Writing to this Register will cause the counter to be reset to its reset value of FFFCh. Flags and enable bits remain unaltered by this operation.

ALTERNATE COUNTER REGISTER
High byte (001Ah)

Read Only
Reset Value: 1111 1111 (FFh)

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

ALTERNATE COUNTER REGISTER
Low byte (001Bh)

Read/Write
Reset Value: 1111 1100 (FCh)

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

Writing to this Register will cause the counter to be reset to its reset value of FFFCh. Flags and enable bits remain unaltered by this operation.

INPUT CAPTURE REGISTER 2, High byte (001Ch)

Read Only
Reset Value: Undefined

7							0
IC2.15	IC2.14	IC2.13	IC2.12	IC2.11	IC2.10	IC2.9	IC2.8

INPUT CAPTURE REGISTER 2, Low byte (001Dh)

Read Only
Reset Value: Undefined

7							0
IC2.7	IC2.6	IC2.5	IC2.4	IC2.3	IC2.2	IC2.1	IC2.0

OUTPUT COMPARE REGISTER 2, High byte (001Eh)

Read/Write
Reset Value: Undefined

7							0
OC2.15	OC2.14	OC2.13	OC2.12	OC2.11	OC2.10	OC2.9	OC2.8

OUTPUT COMPARE REGISTER 2, Low byte (001Fh)

Read/Write
Reset Value: Undefined

7							0
OC2.7	OC2.6	OC2.5	OC2.4	OC2.3	OC2.2	OC2.1	OC2.0

2.4 SYNC PROCESSOR

2.4.1 Introduction

The Sync Processor handles all the management tasks of the video synchronisation signals, and is used with the Timer and software to provide information and status on the video standard and timings.

Separated Horizontal and Vertical Synchronization pulses, provided on the HSYNCI and VSYNCI pins, are accepted, with polarity detection by software. In this case HSYNCO = HSYNCI (with programmable polarity inversion), without any blanking.
Alternatively a composite sync signal (sync pulses only) may be provided on CSYNCI (or the HSYNCI pin), with automatic synchronisation pulse extraction (with polarity detection by software).

Extraction of VSYNCO may be made from a composite signal (OR, XOR or serration combinations of Horizontal and Vertical components).
Note. If the input is a composite signal both VSYNCO and HSYNCO will be extracted (the latter blanked during VSYNCO pulse as far as potential serration pulses are concerned).

Processed sync pulses may be output to external parts of the circuit through the HSYNCO and VSYNCO pins with programmable polarity. An independent programmable-duration back-porch (clamping) output signal (CLMPO) may also be combined externally to extend the Horizontal sync output on HSYNCO. In the case of extraction of HSYNCO and VSYNCO from CSYNCI this signal is suppressed during vertical blanking.

2.4.2 Functional Description

The function of the Sync Processor can be summarized as the 4 following tasks:

- Check the presence of input signals (VSYNCI, HSYNCI and CSYNCI)
- Polarity Detection (VSYNCI, HSYNCI and CSYNCI)
- HSYNCO, VSYNCO Extraction
- Video Standard Discrimination

These tasks are performed by the Sync Processor in close conjunction with the Timer, and user software.

The block diagram of the Sync Processor is shown in Figure 20. This also shows the internal connections to the Timer Input Capture A (IC_A) and Input Capture B (IC_B).

Checking the presence of input signals

The Sync Processor offers two techniques for checking the prescence of signals, a software intensive direct check and an indirect check which is interrupt driven and uses the hardware of the Sync Processor. This second technique thus offers more time for other tasks.

Direct Check

The direct check is made by monitoring the input status directly on the I/O pins. To do this, the internal multiplexors of the Sync Processor are set to allow the direct pass-through of the incoming synchronization signals through to the corresponding output:

HSYNCI => HSYNCO (PD1)

VSYNCI => VSYNCO (PD2)

The corresponding sync output pins are set to the normal I/O mode (by setting the SYNOP bit of PIOCR) and the state of the inputs are read over a period of time in order to detect any transitions on the input. If found it can be assumed that signals are present.

CSYNCI can be monitored by using the PD0 input function directly.

Indirect Check

To use the indirect check, the multiplexers are set to connect the VSYNCI input to Timer Input Capture A and and HSYNCI or CSYNCI to the Input Capture B.

step I - Checking VSYNCI. Any interrupt request coming from IC_A is monitored. (On detecting VSYNCI, the software may either detect the VSYNCI polarity or check for the presence of HSYNCI).

step II - Checking HSYNCI. The input HSYNCI is connected directly to IC_B. An interrupt request is waited for (on detecting HSYNCI, the software may either detect the HSYNCI polarity or check the CSYNCI presence).

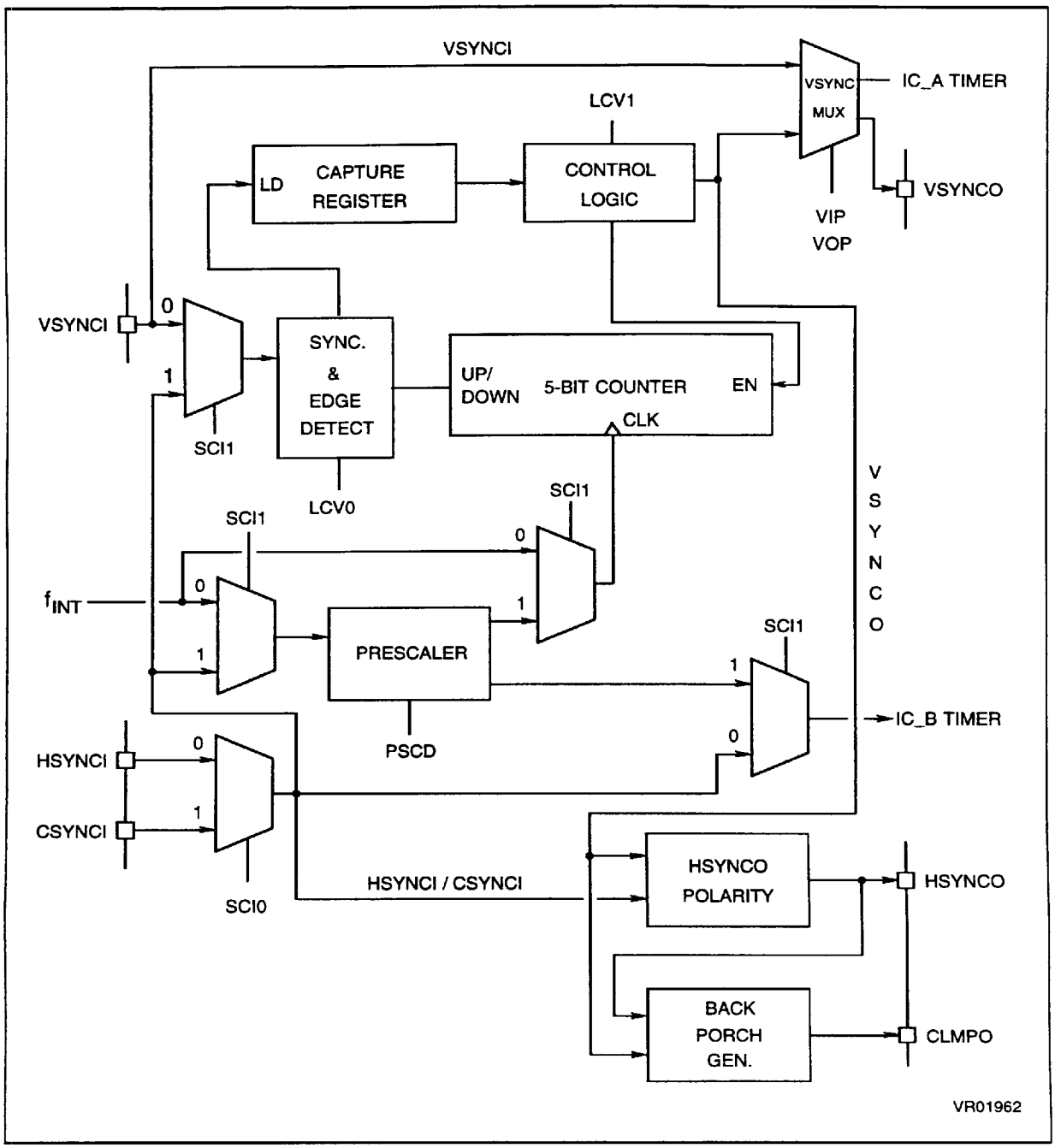
step III - Checking CSYNCI. The CSYNCI input is connected directly to IC_B. An interrupt request is waited for.

Note. Input Capture A edge detection polarity may be selected to be positive or negative.

Input Capture B edge detection polarity is fixed to negative edges.

Steps I-III may be carried out in parallel.

Figure 19. Sync Processor Block Diagram



Polarity Detection.

The Sync Processor again offers two techniques for checking the polarity of signals, a software intensive direct check and an indirect check which uses the hardware of the Sync Processor. This second technique thus offers more time for other tasks.

Indirect Polarity Detection

To check signal polarity with the indirect method, the internal 5-bit up/down counter is used. At the beginning of the detection phase, '11111' is written into the SYNC control register (CVM bits). These bits are updated by the 5-bit counter value at every detected edge (for example: positive) on the signal considered. The counter increments when the signal is high; otherwise it decrements.

Software can thus check the SYNC Processor capture register after an interrupt (with the signal connected to IC_A or IC_B) or by polling. In case of a positive polarity, the capture value will be '00000' as the counter stays at this value after underflowing. Otherwise, it will be different to '00000' (assured by the resolution of the counter) and thus be negative polarity.

This one-shot detection approach covers separate HSYNCI and VSYNCI signals only. In case of a composite incoming signal, the software should consider reading the Sync Processor register every 7 or 8 interrupt requests, or a number of lines to ensure that the readings are outside the vertical blanking period and to minimize the possibility of misleading polarity values.

Direct Polarity detection

The alternative, software intensive, technique is to read the status of the VSYNCI and HSYNCI signals through the I/O pins as summarized in the previous task description.

An average value can be determined by counting (in software) the number of times the level is '1' and the number of times the level is '0' over a period lasting several frame or line periods or by repeating the counting several times over an equal period. The total values can then be compared and the greater of the counts between '1' and '0' indicates the signal polarity.

Figure 20. Horizontal Sync. Input Timing

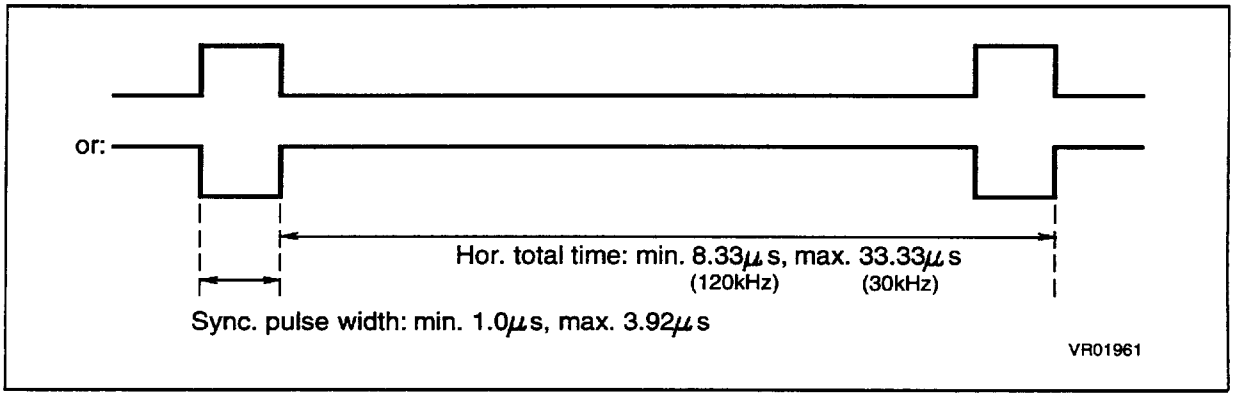
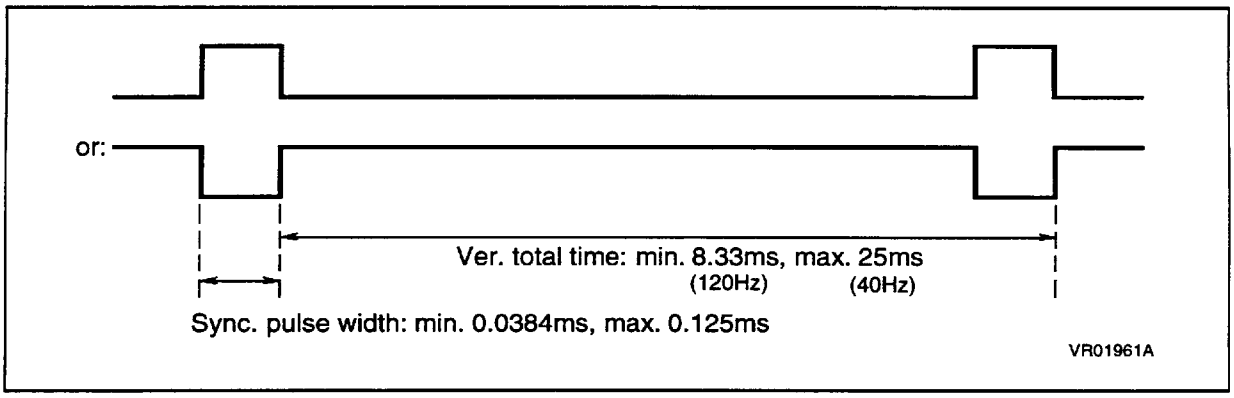


Figure 21. Vertical Sync. Input Timing



VSYNCO Extraction

VSYNCO is extracted from CSYNCI with the aid of the 5-bit up/down counter. Initially, the width of a Horizontal sync component pulse is determined automatically by the hardware, which defines a threshold for the counter (which may be replaced with an optional user defined value including a tolerance factor), the circuit then monitors for any incoming period greater than this captured value. This is then processed as the VSYNCO signal.

The user software should first select the acquisition mode to measure the internal Horizontal sync component pulse width. The time-equivalent value is read after this value is captured in the internal register. If a user-defined tolerance is to be added, then an updated value can be re-written into the register.

The capture occurrence can be indicated by the timer Input Capture interrupt or noted by reading a new value on the Sync Processor Control Register. After this step, the software should set the extraction mode to continue the VSYNCO generation by hardware as shown in the following paragraph for a negative polarity signal.

In extraction mode, the 5-bit comparator checks the counter value with respect to the threshold. When the counter reaches the threshold on its way down, VSYNCO is asserted. During the vertical blanking, counter value is decreased until it reaches a programmable minimum, i.e. it does not underflow. When the vertical period is finished, the counter starts counting up and when the maximum is reached, VSYNCO is negated. The extracted signal may be validated by software since it is input to Timer IC_A.

The threshold is greater than the count for a HSYNCI pulse. Serration pulses during vertical blanking are thus filtered out. Similarly, positive CSYNCI signals are covered by properly selecting the edge sensitivity on the Hsync-width-measurement mode.

Standard Discrimination

Discrimination of video standards is supported by software and the Timer used in conjunction with the Sync Processor. For this purpose, either HSYNCI or CSYNCI is prescaled by a fixed factor of 256.

Note. the prescaler may be bypassed for frequency sychronization signals lower than specified.

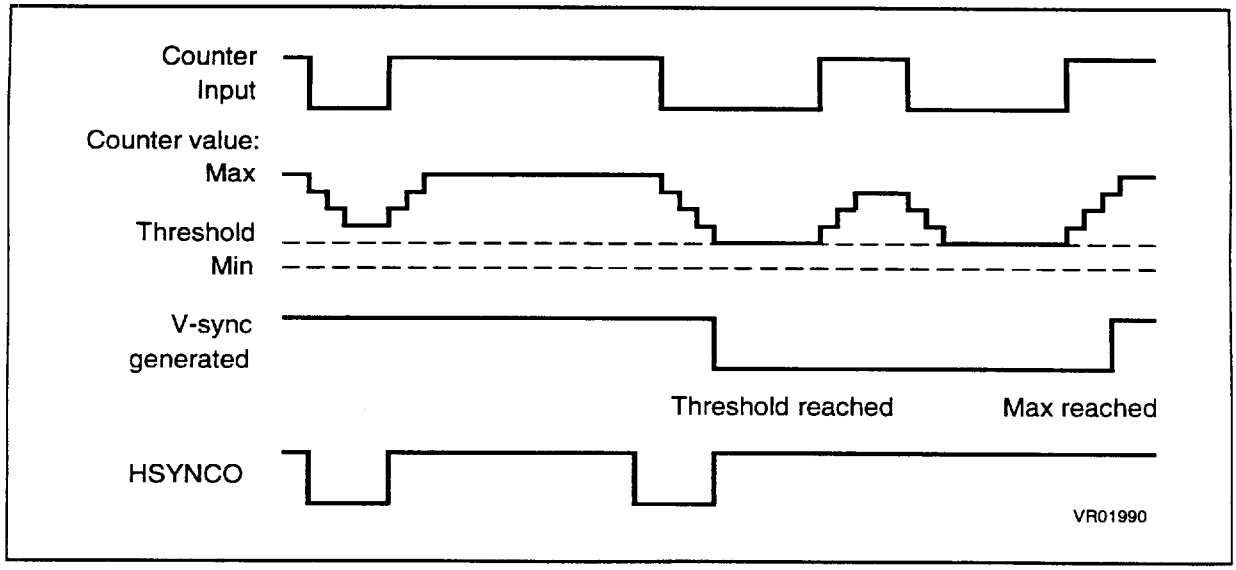
The function is carried out by using the Timer Input-Capture Channels:

a) VSYNCI is directly connected to IC_A (after synchronization).

b) HSYNCI or CSYNCI is prescaled (or not) by a factor of 256 in the Sync Processor and then sent to IC_B.

Signal timing can be directly calculated by the time values between the appropriate interrupts given by the Timer, and then used for comparison against existing pre-defined standards. It is recommended that several captured values are averaged to give a more accurate timing for comparison if serration pulses are present on CSYNCI.

Figure 22. VSYNCO generation for a composite signal (+ serration pulses) with a negative polarity



Inputs

The inputs to the Sync Processor are the Video Synchronization strobe pulses:

VSYNCI (Vertical Sync input, TTL Level, Schmitt triggered).

HSYNCI (Horizontal Sync input, TTL Level, Schmitt triggered).

CSYNCI (Composite Sync input, TTL Level, Schmitt triggered).

NOTE: The Composite Sync signal may also be received on the HSYNCI input if this is supplied by the external circuit and the I/O function of the corresponding I/O pin for CYSYNCI is required.

Input Signal Waveforms.

The input signals must contain only synchronization pulses.

Timing characteristics of HSYNCI and VSYNCI

HSYNCI:

In case of serration pulses on CSYNCI/HSYNCI, these pulses should be externally generated with a minimum half-a-line delay from the VSYNCI edge.

The HSYNCI or CSYNCI signal, optionally prescaled by 256, is connected to the IC_B input (Timer Input Capture B) of the Timer.

The Timer resolution is 500ns for an external oscillator frequency of 8MHz.

Outputs

HSYNCO: HSYNC Output, (CMOS Level). With programmable polarity, this signal is blanked during the vertical period (if the input is a composite signal). Its internal propagation delay has been optimised to its lowest possible delay.

If separated HSYNCI and VSYNCI are provided, no blanking is generated on HSYNCO.

VSYNCO: VSYNC Output, (CMOS Level) with programmable polarity.

VSYNCO is connected to IC_A, input to Timer Input Capture A. The input to Timer is delayed by 125ns-250ns by synchronization with the internal clock.

CLMPO: (CMOS level) back porch clamp signal.

If VSYNCO is extracted from a composite signal, the minimum delay is 500ns + HSYNCO pulse width. The maximum delay is software defined (the threshold value on extraction mode) and corresponds to 8750 ns.

Notes

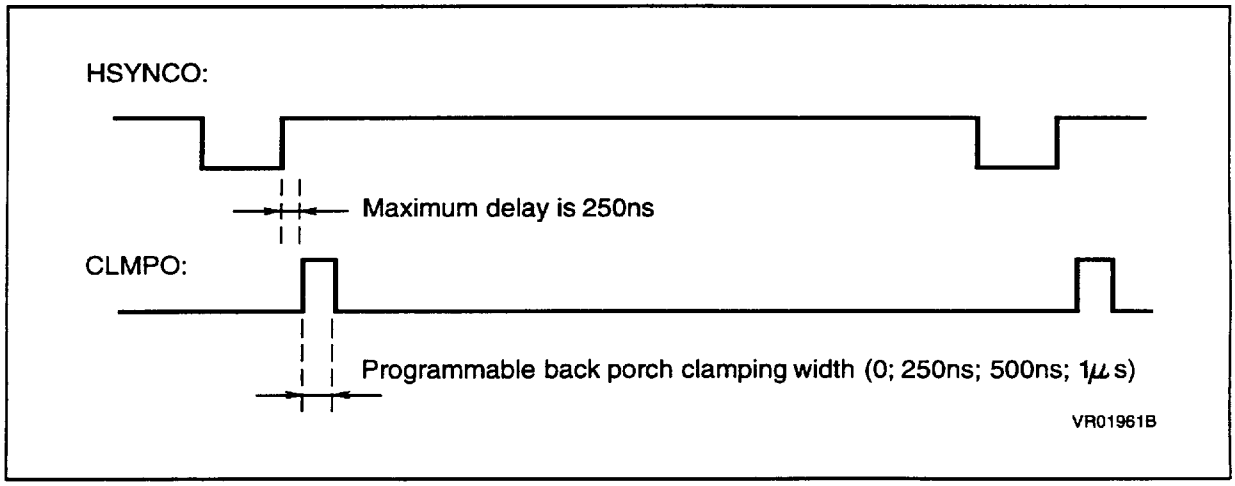
- Standards with less than 256 lines per frame are NOT supported.
- If separated HSYNCI and VSYNCI are provided, no blanking is generated on HSYNCO.
- The following are not supported:

  A back-porch clamp signal generator on VSYNCO.

  A front-porch clamp signal generator.

  Pre/post-equalizing pulses.
- No direct interrupt request is used by the Sync Processor, although the optional interrupt in the timer can be used by the software since VSYNCO and HSYNCI/CSYNCI signals are connected to Input Capture (IC_A and IC_B, respectively).
- The Timer Interrupt Request should be masked during a write access to the SYNC Control Registers.

Figure 23. Back Porch (CLMPO) Delay



2.4.3 Register Description

Two 8-bit read/write registers are used to control the Sync Processor:

- Counter Control Register (CCR)
- Mux Control Register (MCR)

COUNTER CONTROL REGISTER (003Dh)

Reset Value 0000 0000 (00h)

Read/Write

7							0
PSCD	LCV1	LCV0	CV4	CV3	CV2	CV1	CV0

b7 = **PSCD:** Prescaler Disable.

- If set to '0' the 8-bit Prescaler is enabled.
- If set to '1' the 8-bit Prescaler is disabled and preset to 1111 1110b.

b6-5 = **LCV1,LCV0:** VSYNCO Extraction Control Bits

LCV1	LCV0	VSYNC0 Control Bits
0	0	Acquisition mode CSYNCI/HSYNCI Negative polarity Counter capture on input signal falling edge
0	1	Acquisition mode CSYNCI/HSYNCI Positive polarity Counter capture on input signal rising edge
1	0	Extraction mode CSYNCI/HSYNCI Negative polarity CV4-0 = counter minimum threshold
1	1	Extraction mode CSYNCI/HSYNCI Positive polarity CV4-0 = counter maximum threshold

b4-0 = **CV4-0:** Counter Captured Value. These bits correspond to the counter captured value in different modes. Upon VSYNCO extraction, it corresponds to a HSYNCI pulse-width measurement, which may be changed by software before extraction.

MUX CONTROL REGISTER (003Ch)

Reset Value: 0000 0000 (00h)

Read/Write

7							0
BP1	BP0	SCI1	SCI0	HS1	HS0	VOP	VIP

b7-6 = **BP1, 0:** Back Porch (CLMPO) width control

BP1	BP0	Back Porch Width
0	0	No Back Porch
0	1	250ns
1	0	500ns
1	1	1000ns

b5 = **SCI1:** Horizontal/Vertical Signal Path Selection Bit. This bit selects the path for the incoming signals towards IC_A, IC_B, VSYNCO and HSYNCO:

SCI1 = "0" if the task/objective is:

- VSYNCI Polarity Detection using counter
- IC_B connected to HSYNCI/CSYNCI without prescaling

SCI1 = "1" if the task/objective is:

- HSYNCI/CSYNCI Polarity Detection using counter
- HSYNCI/CSYNCI prescaled and connected to IC_B
- VSYNCO extraction

b4 = **SCI0:** HSYNCI/CSYNCI Selection Bit.
This bit selects either HSYNCI/CSYNCI as input.

- If SCI0 = "0" -> HSYNCI pin is selected.
- If SCI0 = "1" -> CSYNCI pin is selected.

b3-2 = **HS1, HS0:** Horizontal Signal Selection Bits. These bits allow inversion of the HSYNCI/CSYNCI polarity, output as HSYNCO, as well as the generation of CLMPO as follows:

HS1	HS0	HSYNC Selection Mode
0	0	CLMPO after HSYNCO rising edge HSYNC0 <- (HSYNCI,CSYNCI)
0	1	CLMPO after $\overline{\text{HSYNCO}}$I rising edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	0	CLMPO after $\overline{\text{HSYNCO}}$I falling edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	1	CLMPO after HSYNCOI falling edge HSYNC0 <- (HSYNCI,CSYNCI)

b1-0 = **VOP,VIP:** Vertical Signal Polarity Selection Bits. These bits are only set/cleared by software and are related to the polarity of the incoming and the outgoing vertical signal.

VIP should be written to '0' by software after detection of a negative polarity on VSYNCI. Otherwise, written to '1'.

VOP should be written to '0' by software to select a negative polarity for VSYNCO. Otherwise, written to '1'.

VOP	VIP	VSYNC Selection Mode
0	0	ICAP_A <- VSYNCI VSYNCO <- VSYNCI
0	1	ICAP_A <- $\overline{\text{VSYNCI}}$ VSYNCO <- VSYNCI
1	0	ICAP_A <- VSYNCI VSYNCO <- $\overline{\text{VSYNCI}}$
1	1	ICAP_A <- $\overline{\text{VSYNCI}}$ VSYNCO <- VSYNCI

2.5 DIGITAL TO ANALOG CONVERTER

PULSE-WIDTH MODULATOR (PWM) + BINARY-RATE MULTIPLIER (BRM)

2.5.1 Introduction

The ST7271 provides two types of Digital to Analog Converters with differing step resolutions based on the Pulse-Width Modulator (PWM) and Binary Rate Multiplier (BRM) Generator technique. These may act as digital potentiometers when used with external filtering to control such elements as brightness, saturation/contrast and other analog variables.

- A 10-Bit PWM/BRM with a repetition rate of 64KHz, 250ns resolution and a step of 5mV (0/5V, excepting DA2). In the 56-pin package, 16 channels are provided with this configuration (channels PWM2-PWM17 with outputs DA2-DA17 respectively). DA2 has a fixed open-drain output, with an external V_{DD} capability up to 12V, while DA3-DA17 are programmable to open-drain or push-pull output configuration, with a 0-V_{DD} (+5V) range.

- A 12-bit PWM/BRM generator (2 Channels: PWM0 and PWM1) with a repetition rate of 64KHz, 250ns resolution and a step of 1.25mV (0/5V excepting DA0). The channels PWM0 and PWM1 correspond with outputs DA0 and DA1 respectively. DA0 has a fixed open-drain output, with an external V_{DD} capability up to 12V, while DA1 is programmable to open-drain or push-pull output configuration, with a 0-V_{DD} (+5V) range.

2.5.2 Functional Description

10-BIT PWM/BRM

The 10-Bits of the 10-bit PWM/BRM are distributed as 6 PWM bits and 4 BRM bits. The generator consists of a 10-bit counter (common for all channels), a comparator and the PWM/BRM generation logic.

PWM Generation

The counter increments continuously, clocked at the Main oscillator frequency divided by 2 (with a period $t_{CLK} = 1/f_{CLK} = 250ns$). Whenever the 6-Least Significant bits of the counter (defined as the PWM counter) overflow, the output level for all active channels is set.
The state of the PWM counter is continuously compared to the PWM binary weight for each channel, software-defined in the relevant PWM register, and when a match occurs the output level for that channel is reset.

This Pulse Width modulated signal is to be filtered with an external RC network, placed as close as possible to the associated pin. This provides an analog voltage proportional to the average charge passed to the external capacitor. Thus for a higher mark/space ratio (High time much greater than Low time) the output voltage is higher. The external components of the RC network should be selected to give the optimum filtering level needed to control the system variable.

Each output may individually have its polarity inverted by software action.

Figure 24. PWM Generation

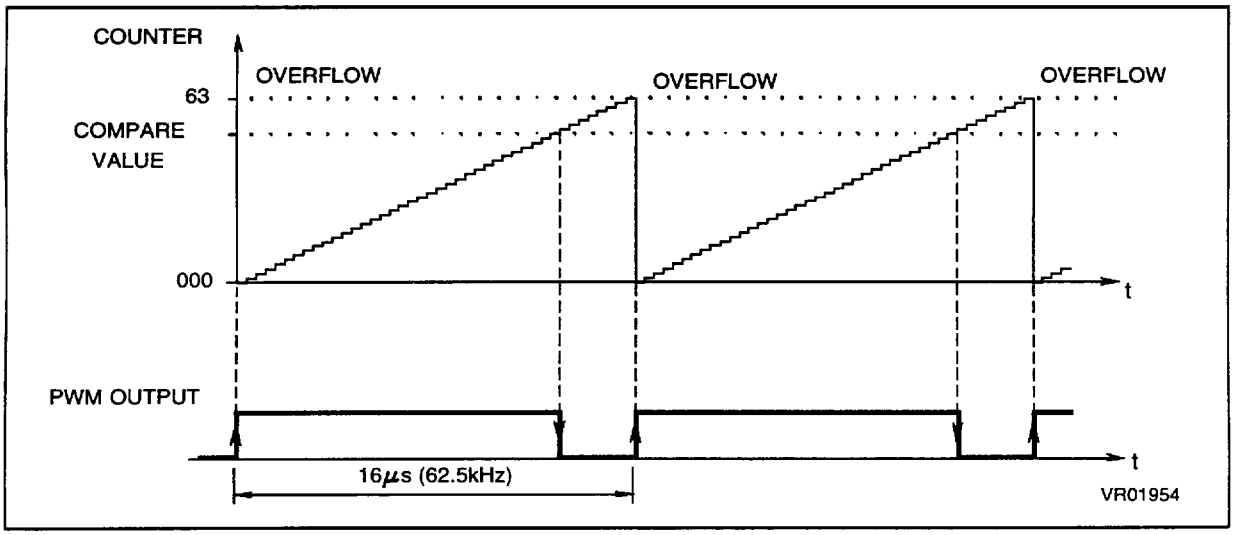
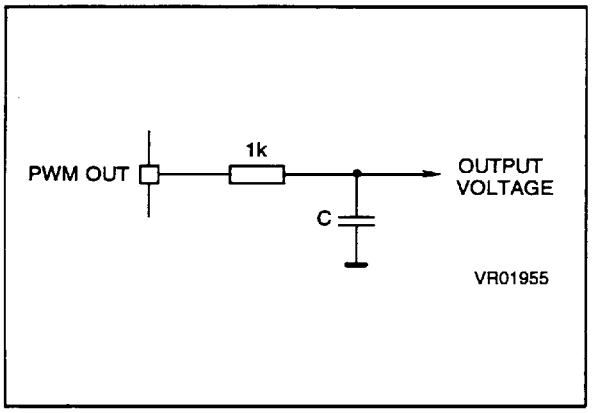


Figure 25. Typical PWM Output Filter



6-Bit PWM Ripple After Filtering

C (µF)	V_{PP_RIPPLE} (mV)	τ (ms)
0.256	78	0.256
2.56	7.8	2.56
25.6	0.78	25.6

Assuming RC filter (R=1kΩ) and V_{DD} = 5V

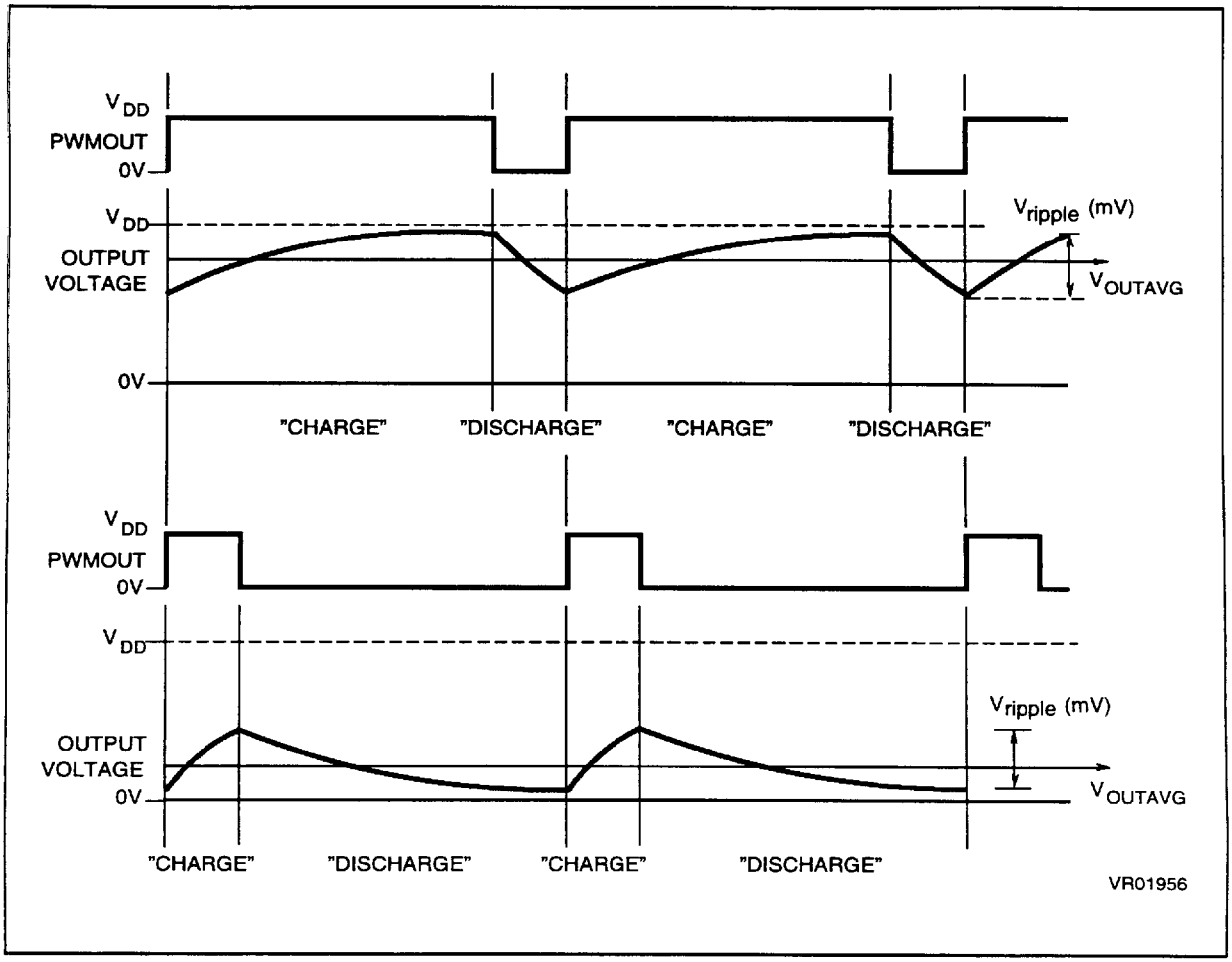
PWM Duty Cycle 50%

Step = 5V/64 = 78mV which requires a minimum

τ (filter time constant) of $\frac{250ns \times 64 \times 64}{4} = 256\mu s$

to ensure integral linearity of ±0.5LSB

Figure 26. PWM Simplified Voltage Output After Filtering (2 examples)



BRM Generation

The BRM bits allow the addition of a 250ns-pulse to widen a standard PWM pulse at specific PWM cycles. This has the effect of "fine-tuning" the PWM Duty cycle (without modifying the base duty cycle), thus, with the external filtering, providing additional fine voltage steps.

The incremental pulses (with duration of t_{CLK} = $1/f_{CLK}$ = 250ns) are added to the beginning of the original PWM pulse. The PWM intervals which are added to are specified in the 4-bit BRM register and are encoded as shown in the following table. The BRM values shown may be combined together to provide a summation of the incremental pulse intervals specified.

The pulse increment corresponds to the PWM resolution. For example, if data 18h is written to the PWM register and data 06h (00000110b) to the BRM register, for a 4 MHz internal clock (250ns resolution), a 6.0 µs-long pulse will be output at every 64µs interval except at those numbered as #2,4,6,10,12,14 where the pulse is broadened to 6.25µs.

Note. If data 00h is written to both PWM and BRM registers, the generator output will remain at "0". Conversely, if both registers hold data 3Fh and 0Fh, respectively, the output will remain at "1" for all intervals #1 ≤ i ≤ #15, but it will return to zero at interval #0 for an amount of time corresponding to the PWM resolution (250ns).
An output can be set to a continuous "1" level by clearing the PWM and BRM values and setting POL = "1" (inverted polarity) in the PWM register. This allows a PWM/BRM channel to be used as an additional I/O pin if the DAC function is not required.

Table 7. 4-Bit BRM Added Pulse Intervals

(interval #0 not selected)

BRM 4-Bit Data	Incremental Pulse Interval #
0000	none
0001	i = 8
0010	i = 4,12
0100	i = 2,6,10,14
1000	i = 1,3,5,7,9,11,13,15

Figure 27. BRM pulse addition (PWM > 0)

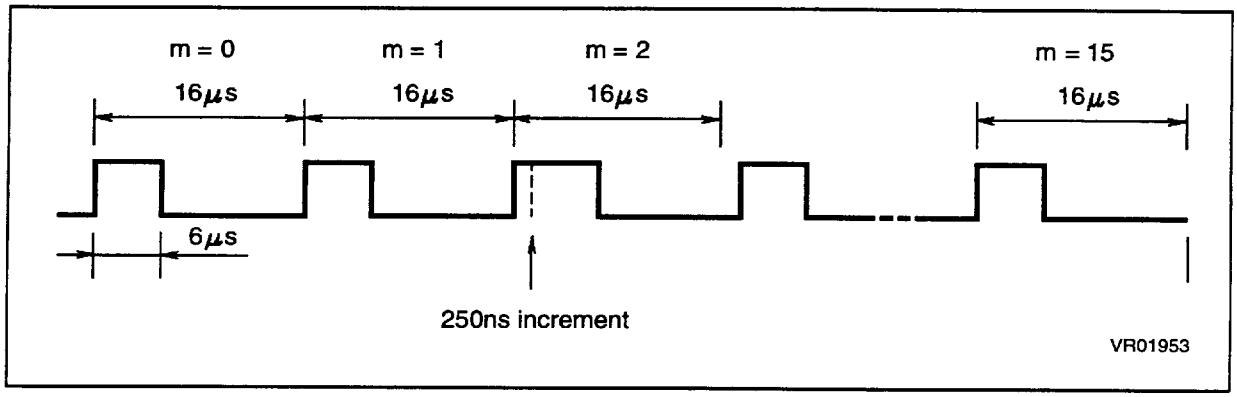


Figure 28. Simplified Filtered Voltage Output Schematic with BRM Added

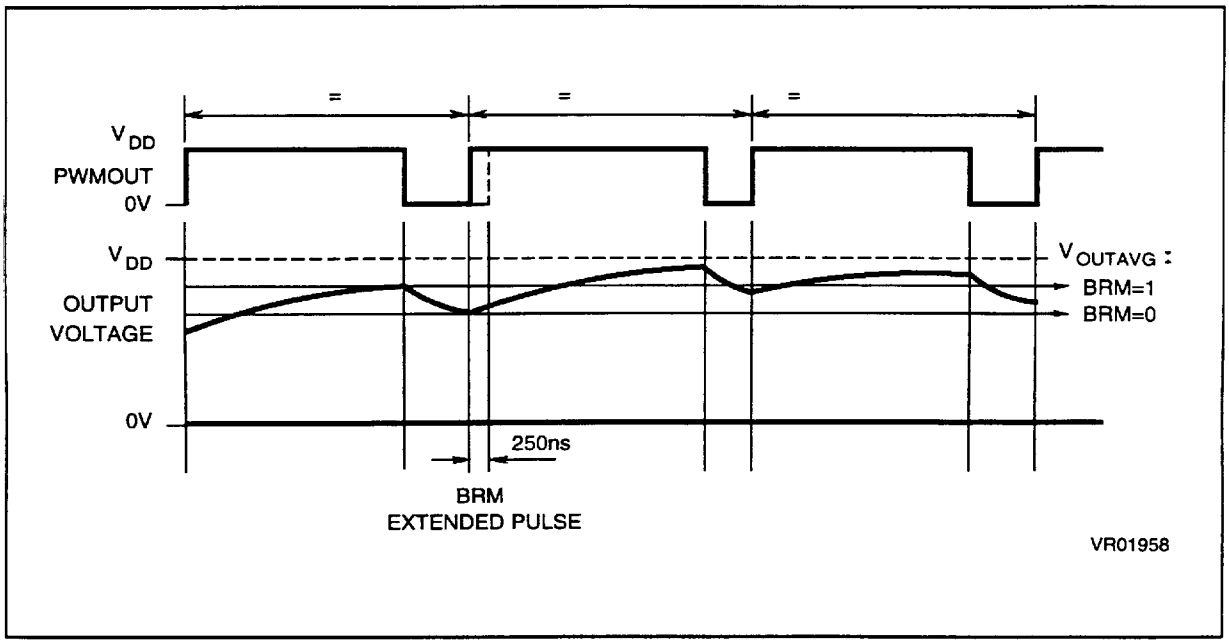
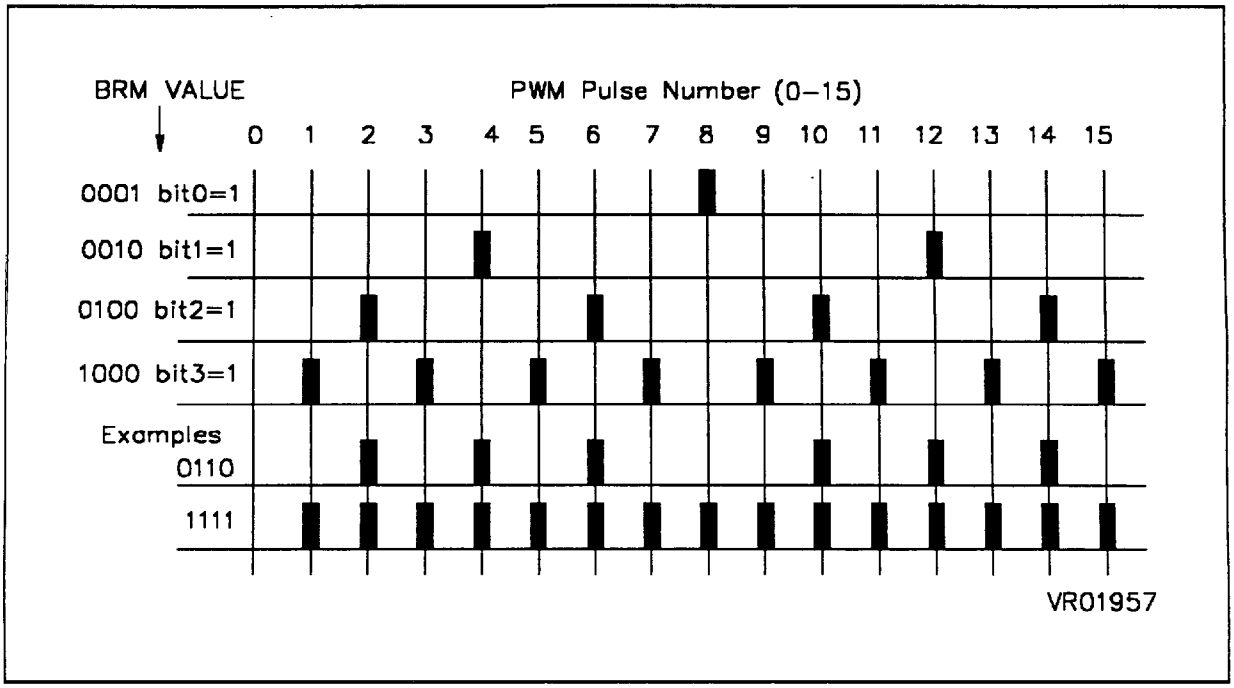


Figure 29. Graphical Representation of 4-Bit BRM added pulse Positions



12-Bit PWM/BRM

The 12 Bits of the two channels of the 12-bit PWM/BRM generator are distributed as 6 PWM bits and 6 BRM bits. The two 12-bit channels correspond to PWM0 and PWM1, and outputs DA0 and DA1 respectively.

PWM Generation

The functionality of the PWM generation is equivalent to the PWM generation of the 10-bit PWM/BRM described in the previous paragraph and so will not be repeated here. Please refer to the previous paragraph for functionality, to be used in conjunction with the following Register description.

BRM Generation

A 6-bit BRM register defining the intervals where an incremental pulse (with duration of t_{CLK} = $1/f_{CLK}$=250ns) is added to the beginning of the original PWM pulse.

Table 8. 6-Bit BRM Added Pulse Intervals

(interval #0 not selected)

BRM 6-Bit Data	Incremental Pulse Interval #
000000	none
000001	i = 32
000010	i = 16, 48
000100	i = 8, 24, 40, 56
001000	i = 4,12, 20, 28, 36, 44, 52, 60
010000	i = 2,6,10,14,...,50, 54, 58, 62
100000	i = 1,3,5,7,9,...., 55, 59, 61, 63

2.5.3 Register Description

10-bit PWM/BRM REGISTERS

On a channel basis, the 10 bits are separated into two data registers:

A 6-bit PWM register corresponding to the binary weight of the PWM pulse.

A 4-bit BRM register defining the intervals where an incremental pulse is added to the beginning of the original PWM pulse. Two BRM channel values share the same register.

PULSE BINARY WEIGHT REGISTER

Register PWMi, i=2,17 (see register map)

Reset Value 1000 0000 (80h)

PWM Pulse Binary Weight

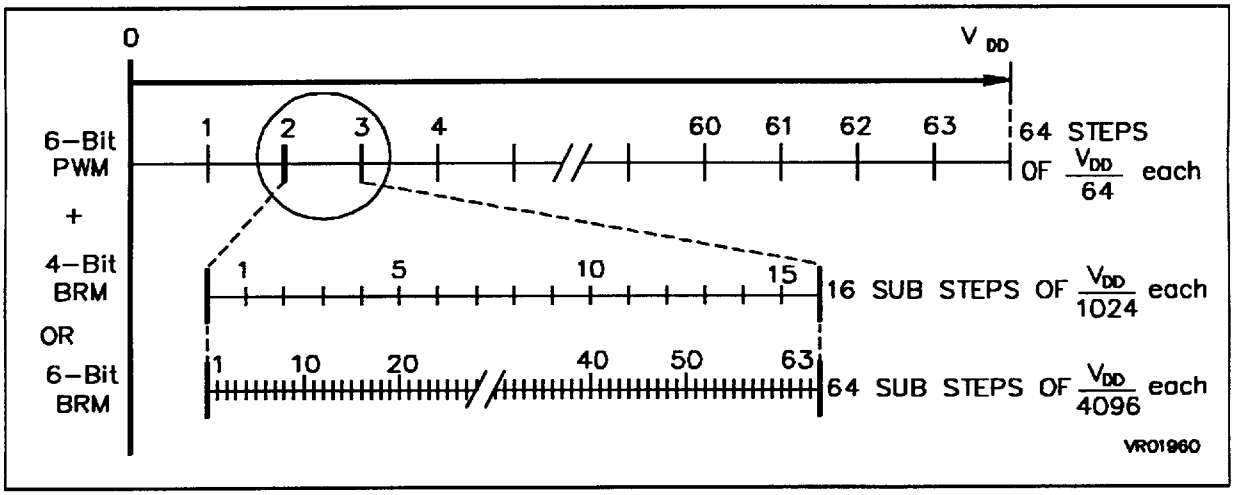
7							0
1	POL	P5	P4	P3	P2	P1	P0

b7 = **Reserved** (read as a "1")

b6 = **POL** Polarity Bit. When POL is set, output signal polarity is inversed; otherwise, no change occurs.

b5-0 = **P5-P0** PWM Pulse Binary Weight for channel i

Figure 30. Precision for PWM/BRM Tuning for V_{OUTEFF} (After Filtering)



DIGITAL TO ANALOG CONVERTER (Continued)

BRM REGISTER

Register BRMi and BRM(i+1), i=2,16

Reset Value: 0000 0000 (00h)

7							0
B7	B6	B5	B4	B3	B2	B1	B0

b7-4 = **B7-B4** BRM Bits (channel i+1)

b3-0 = **B3-B0** BRM Bits (channel i)

12-bit PWM/BRM REGISTERS

For each of the two channels, the 12 bits are separated into two data registers:

A 6-bit PWM register corresponding to the binary weight of the PWM pulse.

A 6-bit BRM register defining the intervals where incremental pulses are added to the beginning of the original PWM pulse.

PULSE BINARY WEIGHT REGISTER

Register PWMi, i=0,1 (see register map)

Reset Value: 1000 0000 (80h)

PWM Pulse Binary Weight

7							0
1	POL	P5	P4	P3	P2	P1	P0

b7 = **Reserved** (read as a "1")

b6 = **POL** Polarity Bit. When POL is set, output signal polarity is inversed; otherwise, no change occurs.

b5-0 = **P5-P0** PWM Pulse Binary Weight for channel i

BRM REGISTER

Register BRMi, i=0,1

Reset Value: 1100 0000 (C0h)

7							0
—	—	B5	B4	B3	B2	B1	B0

b7-6 = **Unused**

b5-0 = **B5-B0** BRM Bits (channel i)

Note : From the programmer's point of view, the PWM and BRM registers can be regarded as being combined to give one data value.

For example :

7				PWM			0		3		BRM	0
O	P	P	P	P	P	P	P	+	B	B	B	B

Effective* DAC value

11			8	7			=				0
O	P	P	P	P	P	P	P	B	B	B	B

* with external RC filtering

PWM/BRM OUTPUTS

The PWM/BRM outputs are assigned to the following pins (unless otherwise stated, they are 10-Bit PWM/BRM, push-pull/open-drain output (0-V_{DD}) configuration):

Table 9. PWM/BRM Pin Assignment

PWM/BRM	Pin	PWM/BRM	Pin
0(1,2)	DA.0	10(3)	DA.10
1(1)	DA.1	11(3)	DA.11
2(2)	DA.2	12(3)	DA.12
3	DA.3	13(3)	DA.13
4	DA.4	14(3)	DA.14
5	DA.5	15(3)	DA.15
6	DA.6	16(3)	DA.16
7	DA.7	17(3)	DA.17
8	DA.8		
9	DA.9		

Notes:

1. 12-Bit PWM/BRM
2. Fixed Open-Drain
3. Not present in 42 pin package

2.6 SERIAL PERIPHERAL INTERFACE

2.6.1 Introduction

The Serial Peripheral Interface (SPI) is an interface which allows several MCUs or peripherals to be interconnected within a single "black box" or on the same printed circuit board.

With the SPI interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and uses a separate signal. An SPI system may be configured as containing one master MCU and several slave MCUs, or as a system in which a MCU is capable of being either a master or a slave.

The SPI features:

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- 2 MHz (maximum) master bit frequency
- 4 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-Master mode fault protection capability.

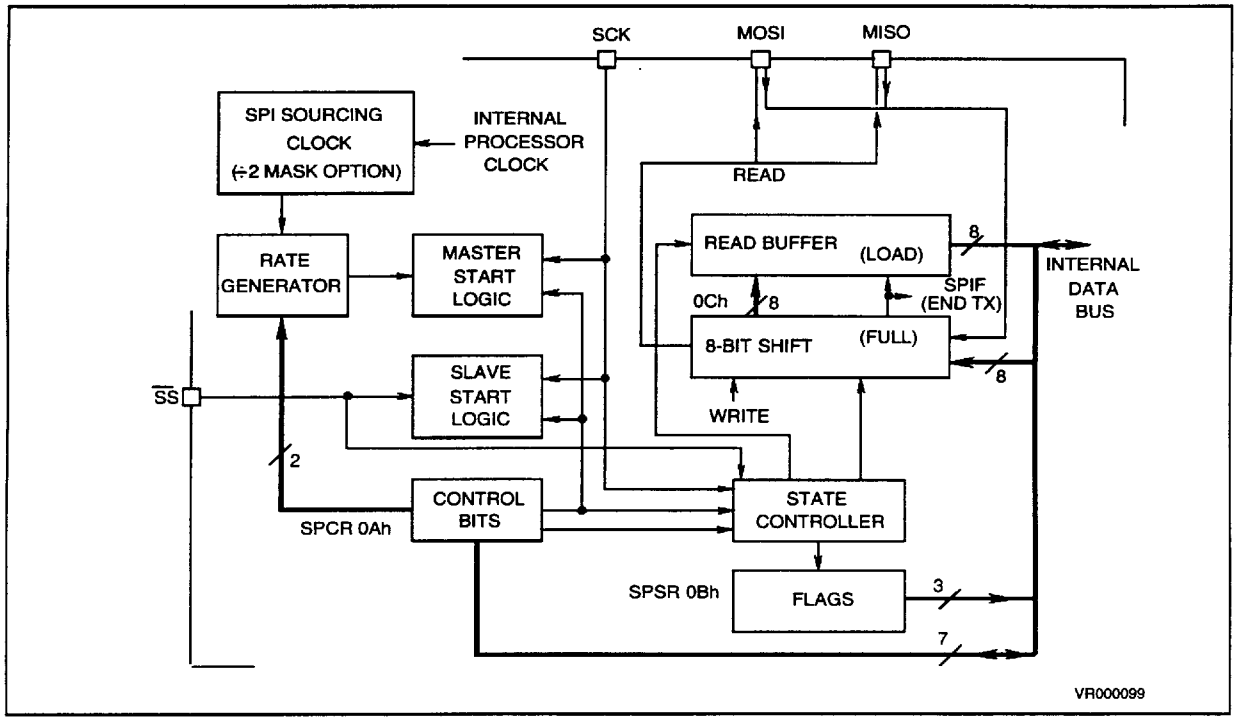
2.6.2 Functional Description

A block diagram of the serial peripheral interface (SPI) is shown in Figure 31. In a master configuration, the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator data register) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the control logic as well as the 8 bit shift register.

As a master device, data is parallel loaded into the 8 bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8 bit shift register. After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{SS}$ pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8 bit shift register.

Figure 31. Serial Peripheral Interface Block Diagram



Note. The user can select by mask option if the internal processor clock signals divided by 2 or not before entering the rate generator.

After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.
During a write cycle, data is parallel loaded into the 8 bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Figure 33 illustrates the MOSI, MISO and SCK master-slave interconnections. Note that the master $\overline{SS}$ pin is tied to a logic high and the slave $\overline{SS}$ pin is tied to a logic low.

2.6.3 Inputs and Outputs

The four basic signals (MOSI, MISO, SCK and $\overline{SS}$) are described in the following paragraphs. These signals are enabled for the corresponding pins of Port C by setting the SPIE bit of the Serial Peripheral control register (if these bits have been configured with a pull-up resistor through the PUPC bit of the Programmable Input/Output Register, then the pull-up is disabled and the port bits use the push-pull output configuration automatically). Each SPI signal function is described for both the master and slave mode.

Master Out Slave In (MOSI)
The MOSI pin is configured as a data output in the master (mode) device and as a data input in the slave (mode) device.

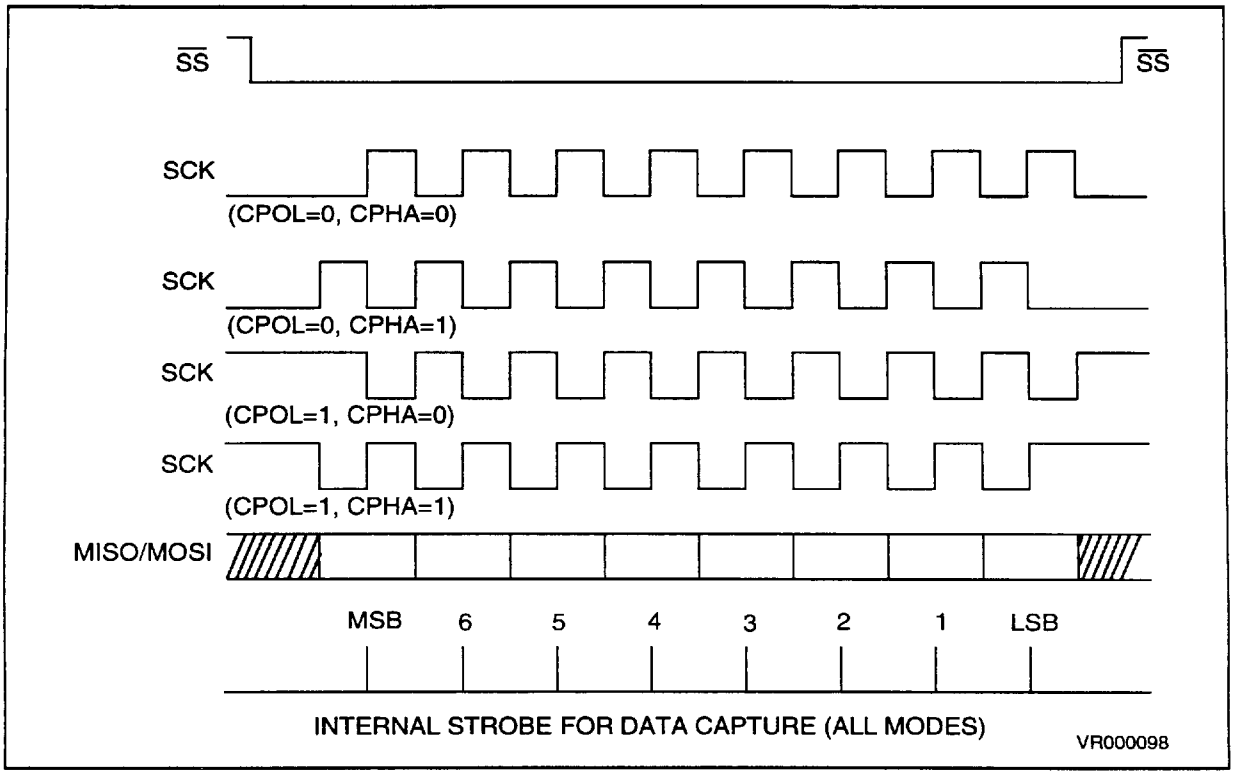
In this manner data is transferred serially from a master to a slave on this line, most significant bit first, least significant bit last. The timing diagram of Figure 32 summarizes the SPI timing diagram shown in the electrical specifications and shows the relationship between data and clock (SCK).

Figure 32 also shows the four possible timing relationships which may be chosen by using the SPI control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line one half-cycle before the clock edge (SCK) to allow the slave device to latch the data.

Both the slave device(s) and a master device must be programmed to similar modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in syn-

Figure 32. Data Clock Timing Diagram



chronized with the same clock signal (which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A status bit (SPIF) in the serial peripheral status register (SPSR), location 0043h, is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location 0042h). When operating as a master, the user should set the MSTR bit to a logic one, giving the MOSI pin as an output.

Master In Slave Out (MISO)
The MISO pin is configured as data input in a master (mode) device and as data output in a slave (mode) device. Data is transferred serially from a slave to a master on this line, most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master, i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of Figure 32 shows the relationship between data and clock (SCK).

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location 0042h) should be to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic 1 level of the $\overline{SS}$ pin, i.e., if $\overline{SS}$ = 1, then the MISO pin is placed in the high-impedance state, whereas if $\overline{SS}$ = 0 the MISO pin is an output for the slave device.

Slave Select ($\overline{SS}$)
The slave select ($\overline{SS}$) pin is an alternate input function on PORT C pin 5 (PC5) enabled by setting the SPE bit of the serial peripheral control register. It receives an active low signal that is generated by a master device to enable slave devices to accept data.

To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle.

Figure 32 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are :

a) CPHA = 1 or 0, the first bit of data is applied to the MISO line for transfer, and,

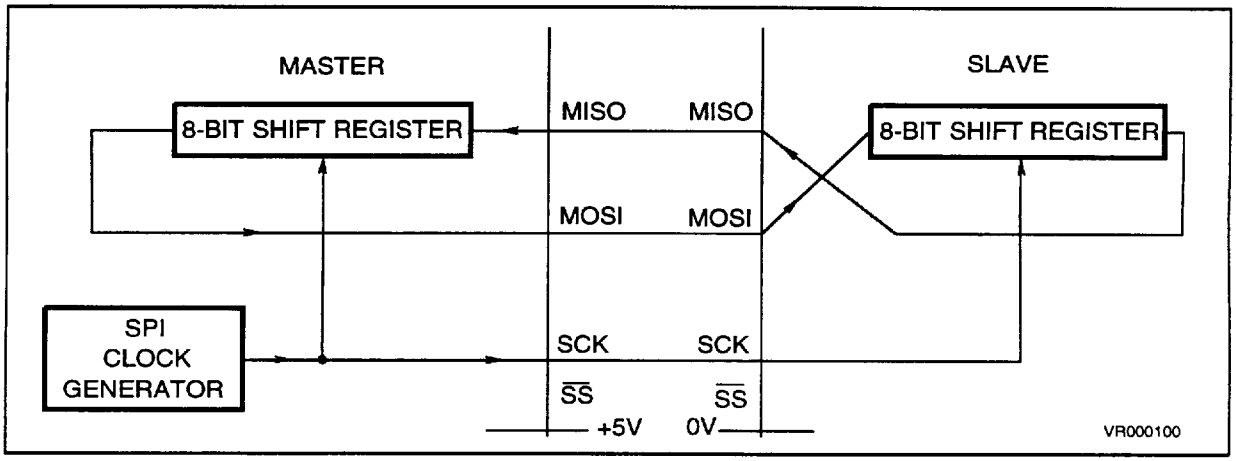
b) when CPHA = 0 the slave device is prevented from writing to its data register.

The state of the $\overline{SS}$ input and of the CPHA bit may have an effect on the I/O data register, refer to following section titled "Collision Detection".

A high level $\overline{SS}$ signal forces the MISO (master in/slave out) line to the high-impedance state. In addition, SCK and the MOSI (master out/slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. The master device will become a slave device any time its $\overline{SS}$ signal input is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system.

Figure 33. SPI Master-Slave Interconnections



When the $\overline{SS}$ line is detected low, the master clears the MSTR control bit (SPCR). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port C SPI pins become inputs). The MODF flag bit in the serial peripheral status register (SPSR) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be master are normally the result of a software error, however, a system could be configured containing a default master which would automatically "takeover" and restart the system.

Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (SPCR) discussed below. Refer to Figure 32 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. A mask option allows the processor clock to be divided by two or not, allowing the input clock to be 4MHz or 2MHz (for 8MHz oscillator).
Two bits (SPR0 and SPR1) in the serial peripheral control register (SPCR) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface.

Collision Detection

An SPI collision is defined as when an attempt was made to write to the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the operation of the MCU.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur both in the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer.

The signal on the $\overline{SS}$ pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero.
When CPHA is logic zero, data is latched with the occurence of the first clock transition. The slave device has no way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the msb of the data onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge.
By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

Detection of write collision is shown in the write collision status bit WCOL in the serial peripheral status register. The WCOL bit is only set if the I/O register is accessed while a transfer is taking place.

Unlike other SPI interfaces, there is no special case of collision undetected by the WCOL bits. The WCOL bit is totally reliable for collision detection.

Since the slave device operates asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurence. This also helps alleviate the user from a strict real-time programming effort.

2.6.4 Register Description

SERIAL PERIPHERAL CONTROL REGISTER (SPCR) (0042h)

Reset condition: 00x0 xxxx

7							0
SPIE	SPE	Res	MSTR	CPOL	CPHA	SPR1	SPR0

b7 = **SPIE:** Interrupt Enable When the serial peripheral interrupts enable bit is high, it allows the occurence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODF) is set to a logic one. It does not inhibit the setting of a status bit.
SPIE is cleared by reset.

b6 = **SPE:** Output Enable Control When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in.
As the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

b5 = **reserved**

b4 = **MSTR** Master/Slave Select The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident.
The MSTR bit is cleared by reset, therefore, the device is always placed in the slave mode during reset.

b3 = **CPOL** Clock Polarity. The clock polarity bit controls the normal or steady state value of the clock when no data is being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred.
CPOL is not affected by reset.

b2 = **CPHA:** Clock Phase The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master or slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relationship. In general the CPHA bit selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge.
CPHA is not affected by reset.

b1,b0 = **SPR1, SPR0:**
These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master. However, these 2 bits have no effect in the slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master.
The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	Internal Processor Clock Divide[1]
0	0	2
0	1	4
1	0	16
1	1	32

Note 1. In this case, the SPI sourcing clock mask option (divider by 2) has not been selected.

SERIAL PERIPHERAL STATUS REGISTER (SPSR) (0043h)

reset condition: 00x0 xxxx

7			6				0
SPIF	WCOL	–	MODF	–	–	–	–

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows :

b7 = **SPIF:**Serial Peripheral Data Transfer Flag
This bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing to its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register.

While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

b6 = **WCOL:** Write Collision Status. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by:

- A read of the serial peripheral data register prior to the SPIF bit being set, or
- A read or write of the serial peripheral data register after the SPIF bit is set.

A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.
WCOL is cleared by reset.

b4 = **MODF:**
The function of the mode fault flag is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set.

The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

- MODF is set and SPI interrupt is generated if SPIE = 1.
- The SPE bit is forced to a logic zero. This blocks all output drive from the device and, disables the SPI system.
- The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register.

To avoid any multi slave conflict in the case of a system of several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of MODF. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

SERIAL PERIPHERAL DATA I/O REGISTER (SPDR) (0040h)
reset condition: undefined

7							0
-	-	-	-	-	-	-	-

The serial peripheral data I/O is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the Write Collision and SPIF status bits to understand the utilization limits of the serial peripheral data I/O register.

2.6.5 Single Master and Multi Master Configurations

There are two types of SPI systems, single master system and multi-master systems.

A typical single master system may be configured, using a MCU as the master and fours MCUs as slaves. The MOSI, MISO and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock,the slave devices all receive it. Since the MCU master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines.

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO lines. For example, three slave devices, enabled for a transfer, are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation.

To ensure that proper data transmission is occuring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

2.7 EAST-WEST PIN CUSHION CORRECTION (EWPCC) GENERATOR

2.7.1 Introduction

The function of the East-West Pin Cushion Correction generator (EWPCC) is to correct the horizontal synchronisation timing signals to compensate for the pin-cushion effect on the display screen, normally caused by the magnetic display components.

The EWPCC operates by storing, during the fabrication process of the monitor, 256 coefficients related to the pin cushion effect in a dedicated memory (EEPROM) and then allowing real-time correction by sending the stored coefficients out through the dedicated D/A Converter (DAC). The analog output is added (off-chip) to the deflection voltage that determines the starting point for the next line on the screen.
These coefficients are uniformly distributed over the effective vertical refresh field of the screen.

Figure 34. Pin-cushion Effect

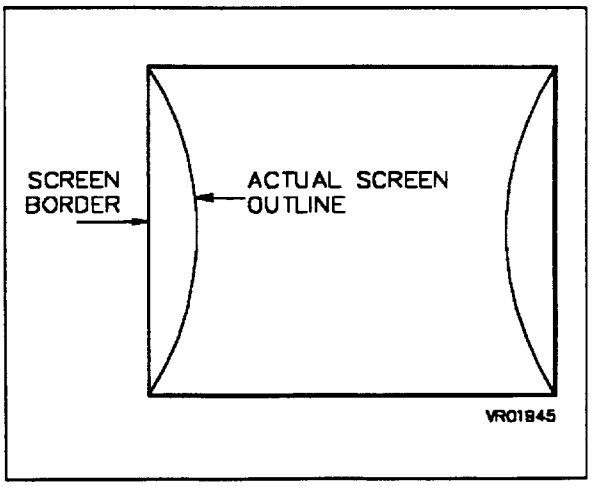
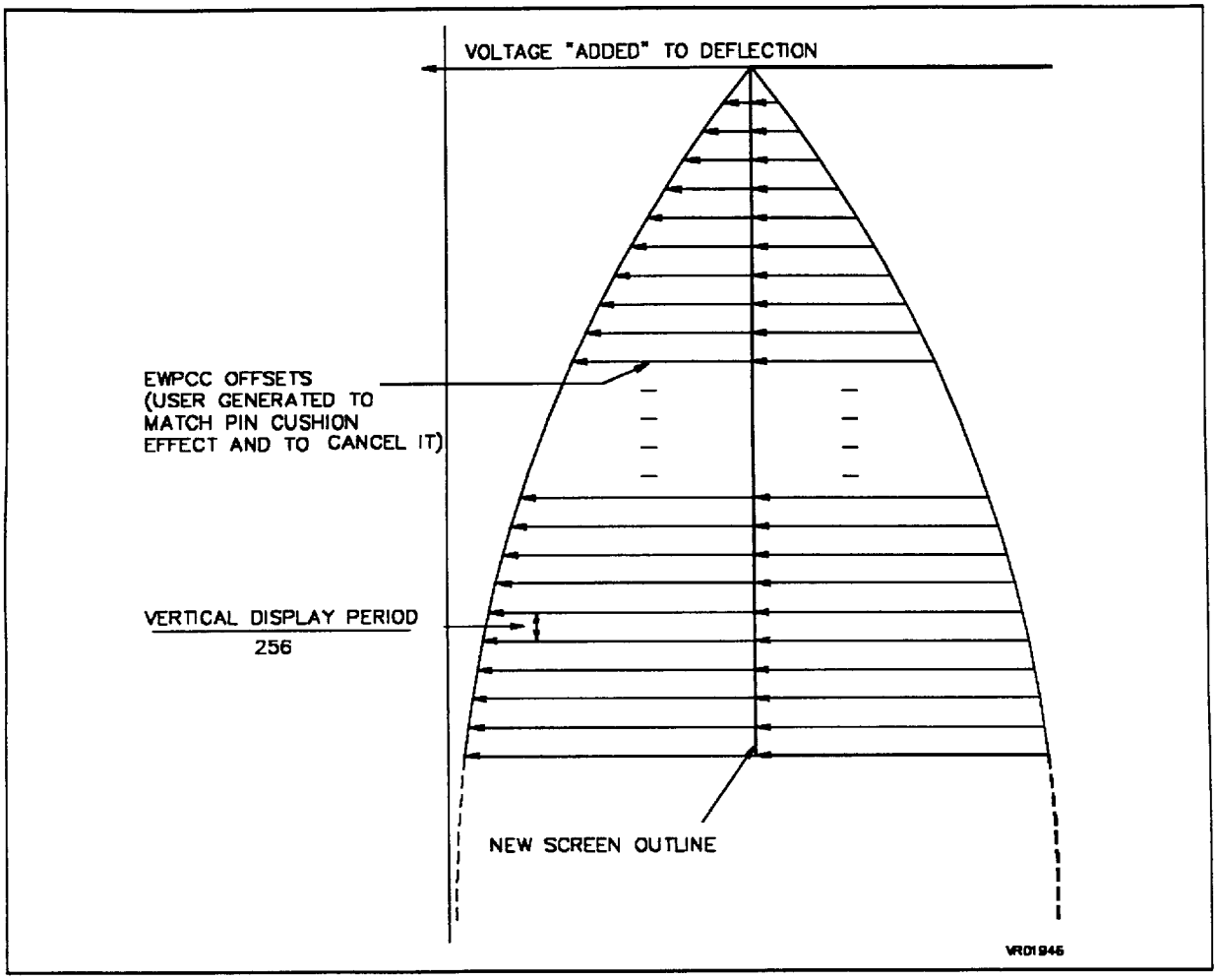


Figure 35. EWPC Coefficients Working on Pin-cushion



2.7.2 Functional Description

The EWPCC output is based on the signal present on the VFBACK input pin. The input signal is required to be asserted during the Vertical Fly-Back period (with a fixed positive polarity assumed).
Warning: when the EWPCC is used, this I/O pin must be set to input mode; if the EWPCC is not used, the pin is available for normal output functions.

For correct operation the frequency of the internal clock of the ST7271 must be 1024 times greater than the flyback frequency.

In normal operation (after initialization), the EWPCC EEPROM is sequentially scanned by the address generator during the time interval that VFback is negated, that is, during the active vertical display period. When VFback is asserted (fly-back period) the address generator is reset, so that the address points to the first EEPROM position and the DAC register is loaded with the first coefficient.

The address generator consists of a 17-bit processor (8-bit up-counter + 9-bit up/down-counter), data/control registers and logic to minimize rounding-up effect.

Figure 37. VFBACK Input Timing

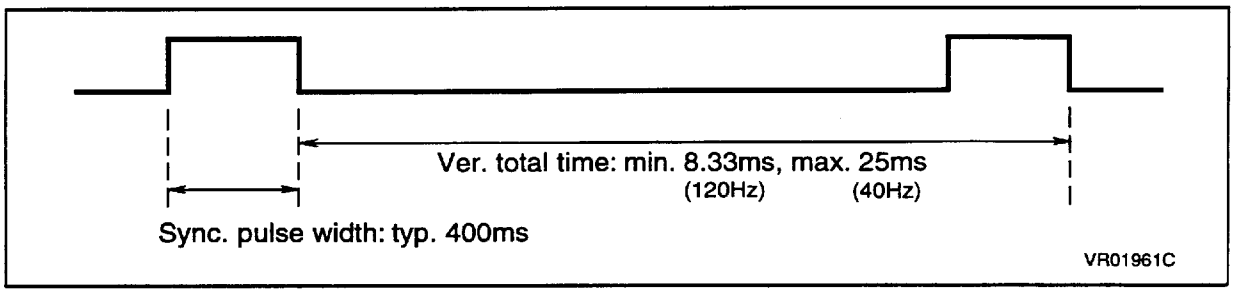
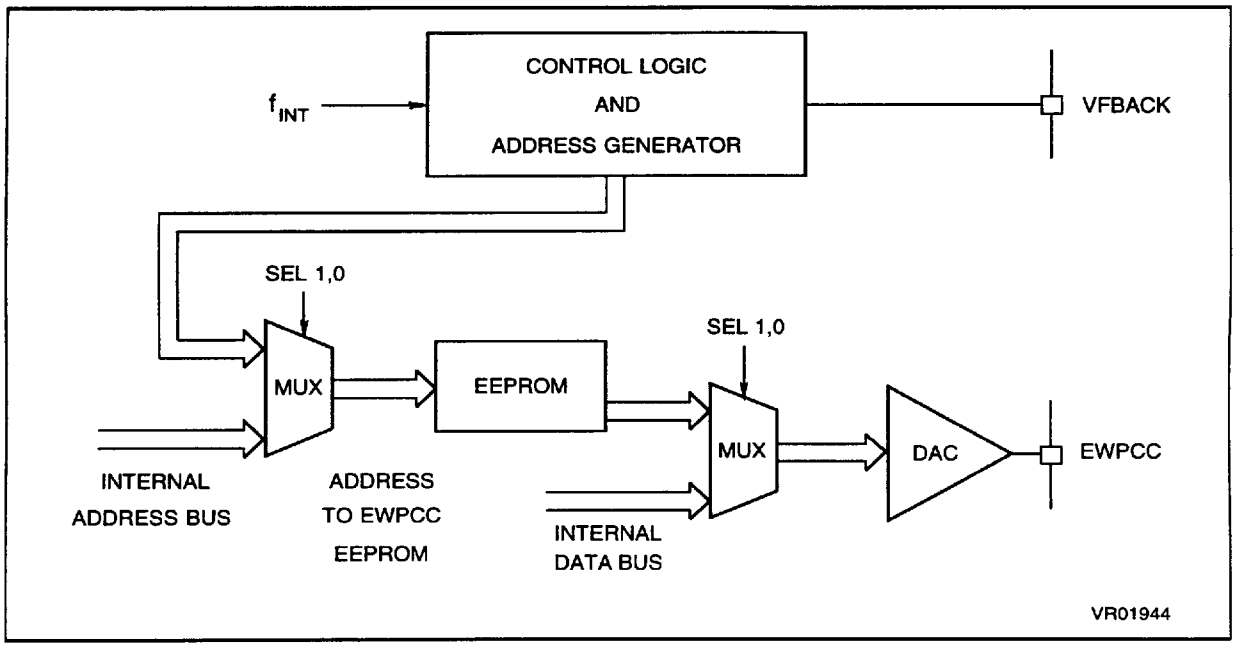


Figure 36. EWPCC Block Diagram



EWPCC Operation

When the EWPCC function is disabled (SEL1,0 = [0,0]), the CPU may fill the EEPROM memory with the EWPCC coefficients calculated or measured during production test or transferred from a preset table within memory.

In normal operation, when enabled, 256 analog levels on the EWPCC output (DAC analog output) are generated in evenly-spaced time/space intervals during the time that VFBACK is negated.
To do so, the EWPCC performs two operating modes that are automatically chained once the sequence is triggered by writing a "1" to the control bit INI in the EWPCC1 register:

a) Acquisition Mode:

The output of the 8-bit prescaler, fed by the CPU clock (f_{CLK}), is counted up by a 9-bit counter during negation of VFback. The rising edge of VFback captures the 9-bit counter value and the 8-bit prescaler MSB into the data registers, as follows:

8-Bit Prescaler MSB -> EWPCC0.0

9-Bit Counter 7 LSB's -> EWPCC0.7,1

9-Bit Counter 2 MSB's -> EWPCC1.1,0

Thus the 9-bit counter contains a value proportional to the duration that VFback is low. As the lower 7 bits of the prescaler are not passed through to the next stage, there is an effective division by 256, so the value in the counter can be regarded as the number of f_{INT} cycles to give the time for the complete non-retrace period equally divided by 256.

The counter and the INI control bit (EWPCC1,2) are then automatically reset and the hardware is switched to the address generation mode.

In order to minimize the truncation effect on the captured value accuracy, the overall 17-bit counter (8-bit prescaler + 9-bit counter) is preset to 00040h during assertion of VFBACK.

Figure 38. Acquisition Mode Block Diagram

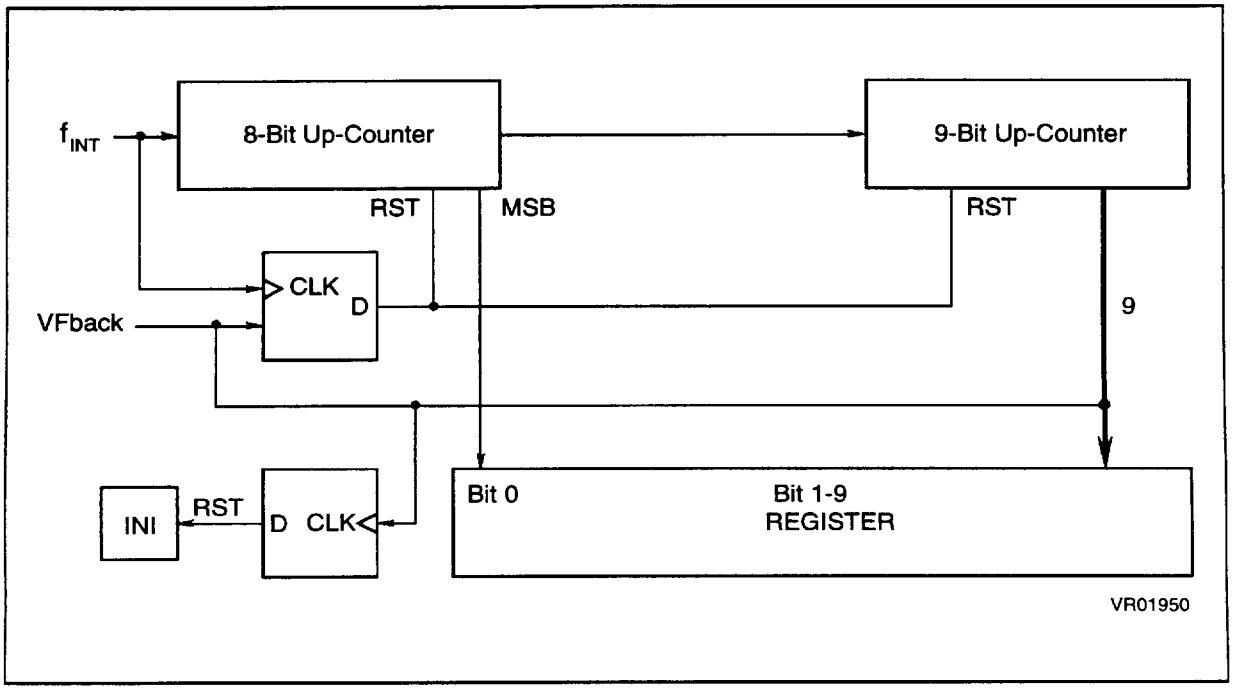


Figure 39. Acquisition Mode Timing

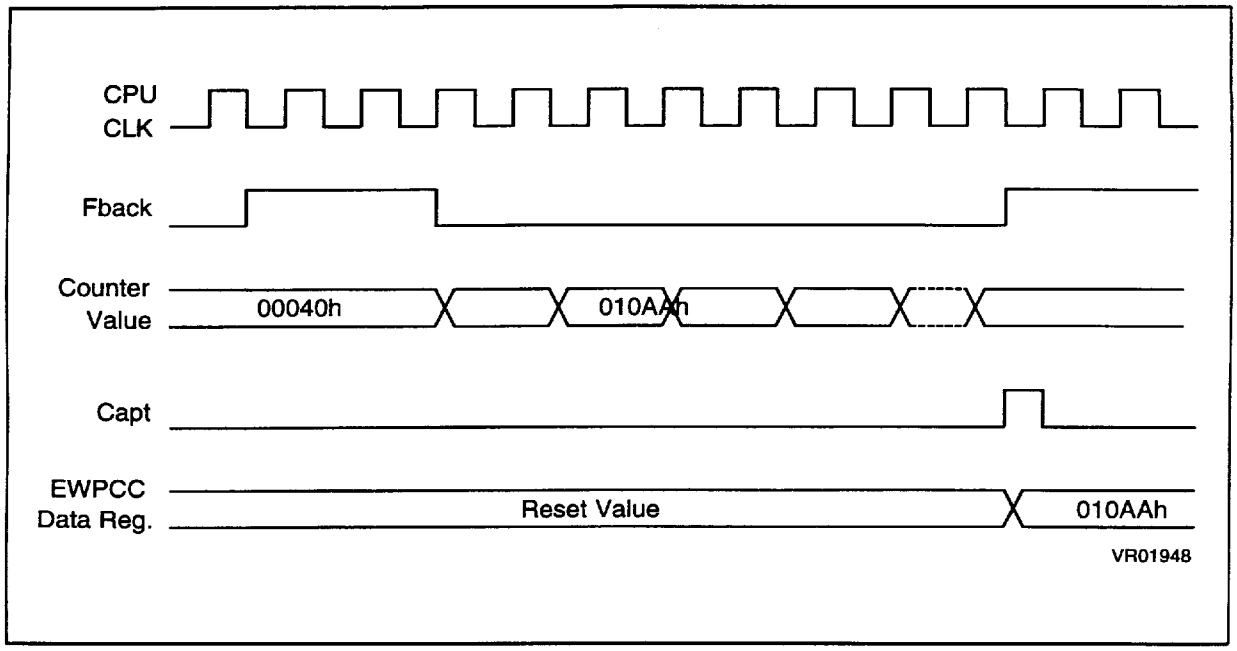
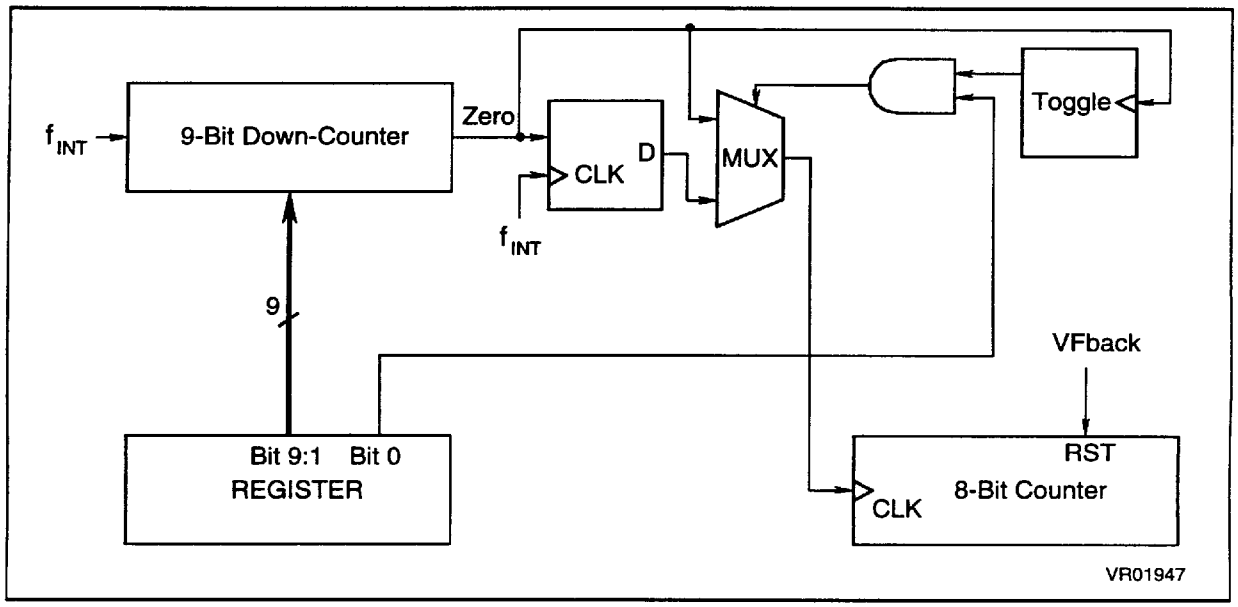


Figure 40. Address Generation Mode Block Diagram



EWPCC GENERATOR (Continued)

Figure 41. Address Clock Generation (Division by N)

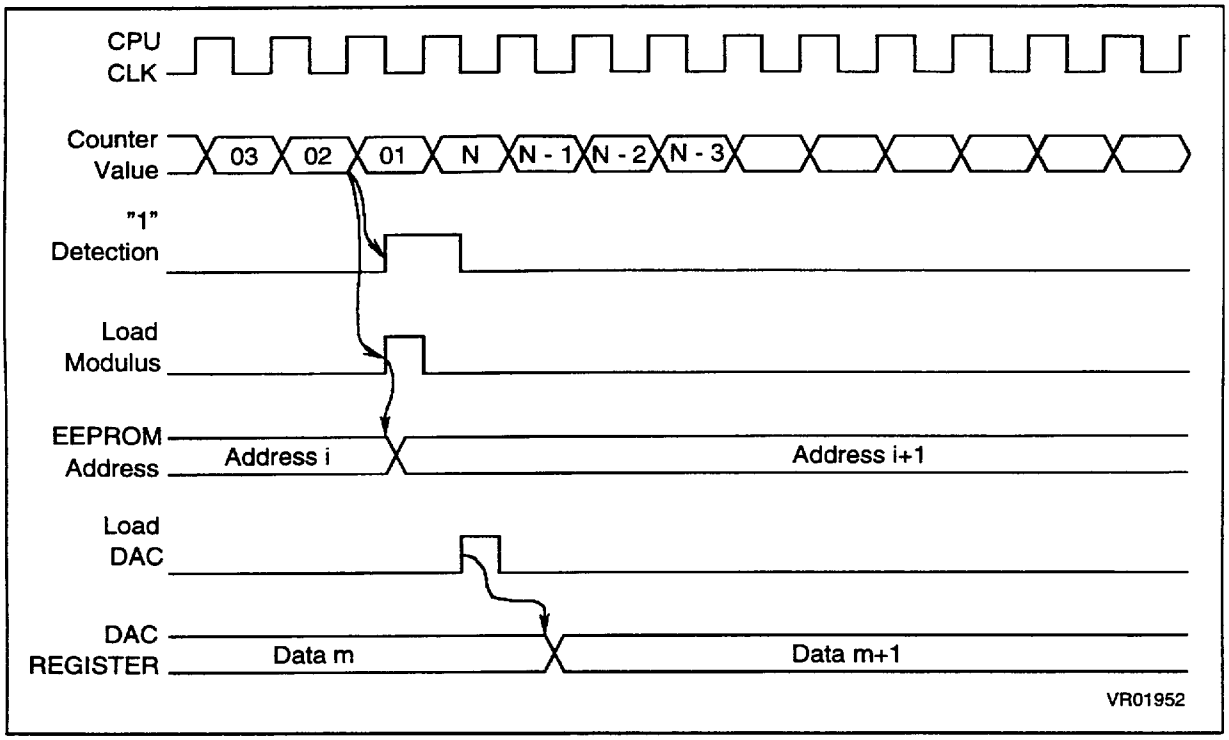
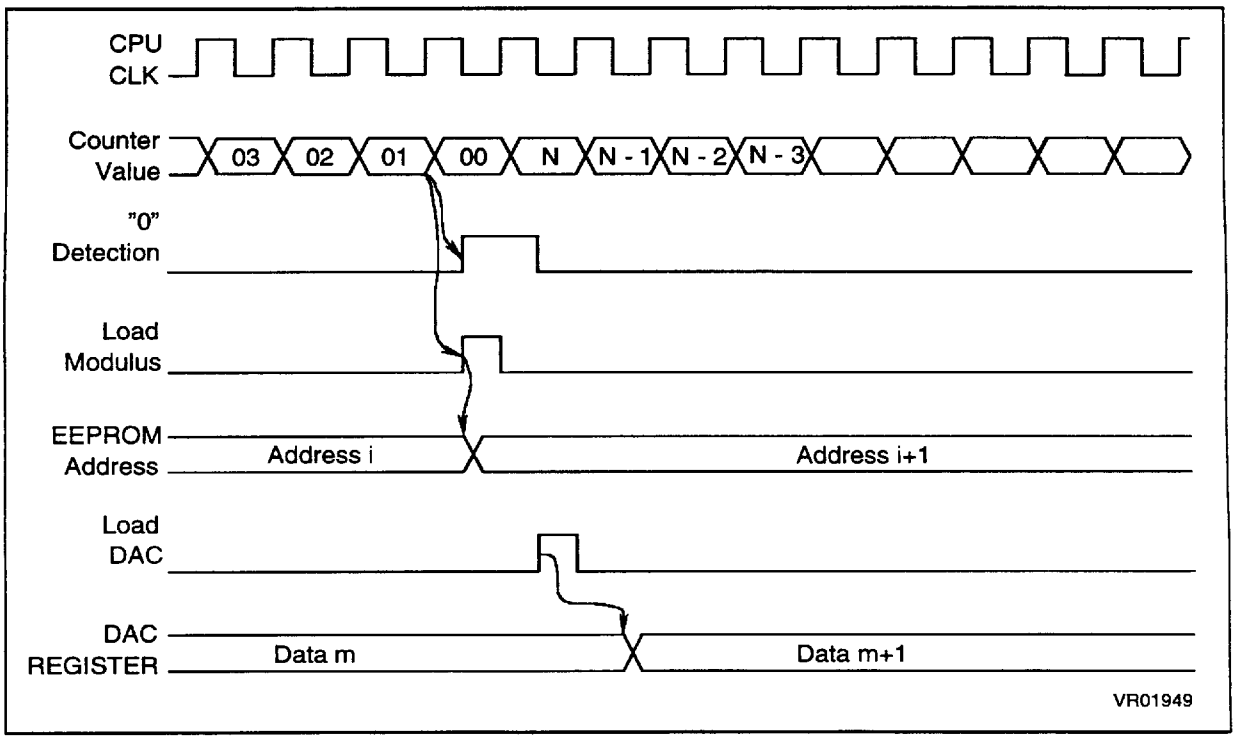


Figure 42. Address Clock Generation (Division by N+1)



b) Address Generation Mode:

Once the acquisition pass is done, addresses to the EEPROM are continuously generated. The 17-bit processor is reconfigured as:

- A 9-bit modulus (divide-by-N/N+1) down-counter, clocked by the CPU clock (f_{CLK}), and reloaded on zero detection with the value captured in the acquisition mode.
- An 8-bit up-counter, clocked by zero detection in the 9-bit counter, which generates the EEPROM addresses.

The division ratio N corresponds to the 9 MSB's of the value captured in the acquisition mode (EWPCC1,0 + EWPCC07,1) which is equivalent to the original number of f_{INT} cycles to achieve the display time of VFBACK low divided by 256. The effect of this is then to divide the VFBACK time evenly by 256 and to increment the address to the EEPROM at each interval.

The choice of N or N+1 is controlled by EWPCC0.0. If this bit is "0", the counter modulus is always N, otherwise an extra step is introduced into the counter to give N+1 and the counter divides by N and N+1 alternately. This algorithm performs a division by (N + 0.5).

Note. After reset, the EWPCC function is disabled and the INI bit is cleared. By writing SEL1,0 = [1,1] and INI=1, the EWPCC function is enabled and the acquisition mode starts.

2.7.3 EWPCC Operation Modes

The EEPROM content addressed by the address counter is loaded into the Digital to Analog converter (DAC) 8-bit data register through a multiplexer and then sent to the D/A Converter. The other input to the multiplexer is the internal Data Bus.

This architecture allows the user the alternative of different configurations, selected by the control bits SEL1, 0.

- SEL1, 0 = [1,1]:

EWPCC Function Enabled (Normal Operation): EEPROM is addressed by the EWPCC Address Sequencer and its values directly sent to the DAC register through a dedicated bus.

- SEL1,0 = [0,1]:

EWPCC Function Disabled and Direct Loading (by a Single Instruction) to the DAC Register from the EEPROM through the dedicated bus. This direct transfer is synchronized with a dummy read instruction of the EEPROM with the addresses generated by the CPU. In this dummy operation, EEPROM data is not loaded onto the main data bus, but can be accessed by reading the DAC register.

- SEL1,0 = [0,0]

EWPCC Function Disabled. EEPROM and DAC in Stand-Alone Mode.

Whenever the dedicated path between the EEPROM and the Digital to Analog Converter is enabled, after accessing the EEPROM, the byte read is sent to the DAC data register. If the DAC is configured as stand-alone, its data register is directly loaded by the CPU through the internal data bus.

Tied to V_{DDA} (Analog VDD, 8V ± 8%) and V_{SSA} (Analog Ground), the DAC is monotonically crescent as it is constructed by a resistive ladder with 256 levels (V_{STEP} = 15.625mV) - between 2V (00h) and 5.9843V (FFh) - output to an unit-gain amplifier and then output to the dedicated pin (EWPCC).

The simulated electrical characteristics are listed in the table below, for a 15pF load capacitance and a 0-70°C temperature range.

DAC Electrical Characteristics

Variable	Min	Typ	Max
Output Current (µA)	340	1050	2200
Response Time (µs) (1)	3.0	3.6	8.0

Note 1. From the moment the new value is loaded onto the DAC register to 90% of the output steady value, for a 2V-step variation.

The architecture of the EWPCC also gives the user the possibility (if necessary) to adjust by software the captured value on the generation mode by writing into the EWPCC registers the new modulus that will be transferred to the counter upon the following underflow condition.

The EWPCC function may also be used for the generation of other waveforms by suitable programming of the EWPCC EEPROM.

Tolerance

The truncation error on the acquired value corresponds to -0.25 ≤ Error ≤ 0.25 (with the algorithm used) which can be related to a maximum shift of 32µm on the vertical axis of the screen, at the 256th address (or a maximum 16µm at the 128th) with respect to the ideal case of zero truncation error when generating evenly-spaced intervals by dividing the time interval in which VFBACK is negated by 256.

2.7.4 Register Description

EWPCC0 REGISTER (000Dh)

Reset Value: 0000 0000 (00h)

7							0
M7	M6	M5	M4	M3	M2	M1	M0

b7-1 = **M7-M1:** 9-Bit Counter 7 LSBs. These bits correspond to the 7 least-significant bits of the captured value from the 9-bit counter when in acquisition mode.

b0 = **M0:** 8-Bit Prescaler MSB. This bit corresponds to the most significant bit of the 8-bit prescaler, captured during acquisition mode.

EWPCC1 REGISTER (000Eh)

Reset Value: 1100 0000 (C0h)

7							0
1	1	0	SEL1	SEL0	INI	M1	M0

b7-6 = **Reserved** (read as "1")

b5 = **Reserved** MUST BE PROGRAMMED AS "0"

b4-3 = **SEL1,SEL0:** Mode Selection Bits. These read/write bits allow selection of configuration, as follows:

SEL1	SEL0	
0	0	EWPCC Function Disabled EEPROM and DAC Stand Alone
0	1	EWPCC Function Disabled EEPROM addressed by CPU and its data sent directly to DAC
1	0	Reserved
1	1	EWPCC Function Enabled EEPROM addressed by EWPCC block Data sent to DAC Register

b2 = **INI:** Acquisition Mode Initialization/Status Bit. Writing a "1" to INI triggers the acquisition mode, after detection of VFback rising edge. During this mode, INI is read as a "1". Once acquisition is completed, this bit is cleared by hardware and address generation mode is chained.

b1-0 = **M1-M0:** 9-bit Counter Value MSB's. These bits correspond to the two most-significant bits of the captured value from the 9-bit counter on acquisition mode.

Notes.

When EWPCC mode is selected (SEL1,0 = [1,1]), any writing operation into the EEPROM control register is blocked by hardware.

- When SEL0 = "1", the EEPROM value is not transferred onto the databus.
- During acquisition mode, re-writing the INI control/status bit as "1" will not have any effect - the acquisition procedure already in progress will not be reset.

DAC REGISTER (003Eh)

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

b7-0 = **D7-D0:** DAC Input Binary Byte. This byte corresponds to the binary value to be converted onto an analog signal.

EWPCC EEPROM CONTROL REGISTER (000Fh)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
Res	Res	Res	—	—	E2ERA	E2LAT	E2PGM

This register contains the bits required to read, erase and program the EWPCC EEPROM. They are defined as follow:

b7-5 = **Reserved**, must be held to "0"

b4,3 = **Unused**

b2 = **E2ERA:** EEPROM Erase. E2ERA must be set to "1" for an erase operation. It must be set after or at the same time as E2LAT. It cannot be changed once an EEPROM address is selected. It is held low when E2LAT is low. It is therefore automatically reset when E2LAT is reset.

b1 = **E2LAT:** EEPROM Latch Enable. When E2LAT is reset to "0", data can be read from the EEPROM. When it is set to "1" and E2PGM reset to "0", a write into the EEPROM array causes the data to be latched, according to the address into one of 8 data registers. An additional internal flag is latched to select the row. The selected columns and row determine the locations involved in the next erase or programming operation. E2LAT must be cleared after each programming or erase operation. E2ERA and E2PGM are forced low when E2LAT is low.

b0 = **E2PGM:** EEPROM Program Mode. This bit allows the internal charge pump to be switched on or off. When set to "1", the charge pump generator is on and the high voltage is applied to the EEPROM array. When low, the charge pump generator is off. E2PGM can only be reset by resetting E2LAT.

2.8 A/D CONVERTER

The Analog to Digital converter is a single 8-bit successive approximation converter. Analog voltages from external sources are input to the converter through up to 8 pins (in the 42-pin DIL package, the selection is made from only 4 pins). The result of the conversion is stored in the 8-bit Result Data Register. The A/D converter is controlled through the A/D Converter Control/Status Register.

2.8.1 Functional Description

The A/D converter is switched on by setting the A/D Converter ON bit (ADON) of the A/D Converter Control/Status Register. A delay time is then required for the converter to stabilize (see characterization section).

When the A/D function is on, pins PB0/AIN0- PB7/AIN7 (pins PB0/AIN0- PB3/AIN3 in the 42-pin package) can be used as analog inputs. The inputs must first be enabled for analog input by setting the corresponding bit(s) of the Port B Configuration Register as described in the Section on I/O Ports. Bits CH3-CH0 of the A/D Converter Control/Status Register (at address 0009h) then select the channel to be converted. The high and low level reference voltages are connected to pins V_{DD} and V_{SS}.

When switched on, the A/D converter performs a continuous conversion of the selected channel. When a conversion is completed (16µs for f_{INT} = 4 MHz), the results is loaded into the read only Result Data Register (at address 0008h) and the COCO (Conversion Complete) flag is set. No interrupt is generated. Any write to the A/D Converter Control/Status Register aborts the current conversion, resets the COCO flag and starts a new conversion.

The A/D converter is ratiometric. An input voltage equal or greater than V_{DD} converts to FFh (full scale) without overflow indication if greater. An input voltage equal to V_{SS} converts to 00h. The conversion is monotonic: the results never decrease if the analog input does not and never increase if the analog input does not.

Using a pin or pins as analog inputs does not affect the ability to read port B as logic inputs.

The 8-bit conversion is accurate to within ± 2 LSB. The minimal conversion time is 16ms (A/D clock frequency at 2 MHz). The A/D converter clock is generated from the CPU clock divided by 2.

The A/D converter can be switched off by resetting the ADON bit. This feature allows the reduction of power consumption when no conversion is in progress. The A/D converter is disabled after power-on and external resets.

The A/D converter is not affected by WAIT mode but, in power sensitive applications, it can be switched off before entering this mode. When the MCU enters the STOP mode with the A/D converter enabled, the A/D clocks are stopped and the converter is disabled until the STOP mode is exited and the startup delay has elapsed. A stabilization time is also required before accurate conversions can be performed.

The reference supply of the converter uses the power supply lines V_{DD} and V_{SS}. The accuracy of the conversion may therefore be degraded by voltage drops in the heavily loaded power lines.

2.8.2 Register Description

A/D CONTROL/STATUS REGISTER (0009h)

Reset condition: 0000 0000 (00h)

7							0
COCO	0	ADON	—	0	CH2	CH1	CH0

b7 = **COCO:** Conversion Complete. COCO is set as soon as a new conversion can be read from the Result Data Register. COCO is cleared by reading the result or writing to the A/D Converter Control/Status Register.

b6 = **Reserved**, must be programmed to 0

b5 = **ADON:** A/D converter On. ADON allows the A/D converter to be switched on and off in order to reduce consumption when needed. When turned on (ADON = 1), a delay time is necessary for the current to stabilize. Conversions can be inaccurate during this time.

b4 = **Not used**

b3 = **Reserved**, must be programmed to 0

b2-b0 = **CH2-CH0:** Channel Selection These bits select the analog input to convert .

CH2-CH0	Pin	Channel
0 0 0	PB0	AIN0
0 0 1	PB1	AIN1
0 1 0	PB2	AIN2
0 1 1	PB3	AIN3
1 0 0	PB4	AIN4
1 0 1	PB5	AIN5
1 1 0	PB6	AIN6
1 1 1	PB7	AIN7

A/D DATA REGISTER (0008h)

reset condition: undefined

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

b7-b0 = **AD7-AD0:** Analog Converted Value.

PAGE(S) INTENTIONALLY BLANK

3.1SOFTWARE DESCRIPTION

3.1.1 Instruction Set

The ST7 instruction set is an 8 bit industry standard instruction set that can be divided into five major groups. All instructions of each group have the same addressing modes.
Refer to ST7 MACRO ASSEMBLER USER'S GUIDE and ST7 PROGRAMMING MANUAL for detailed information.

Group 1 : Register/Memory And Absolute Jump Group In this group most instructions contain two operands. One operand is inherently defined as either the accumulator or an index register. The other operand is fetched from memory using one of the allowed addressing modes. The absolute jump instructions are included in this group because they can use most of the addressing modes of the register/memory instructions.

Examples: LD <ea>, a. This means that the memory byte located at address <ea> is loaded with the 8-bit content of the accumulator A.

The list of the instructions of this group is given in Table 10.

Group 2 : Read - Modify - Write Group These instructions read a register or a memory location, modify its content and write the new value back.

Example : RRC <ea>. This means that the content of the memory byte located at address <ea> is rotated right through the carry bit, the result is written in the memory <ea> and the carry bit.

The list of the instructions of this group is given in Table 11.

Group 3 : Bit Manipulation And Test Group Bit manipulation instructions can set or clear any bit within the first 256 memory locations, except for ROM (020h - 04F) and read-only registers located at addresses 03h, 0Bh, 010h, 013h, 014h and 015h.

Example: BSET <ea>, #b. This sets the bit #b of memory location <ea>.

Test instructions can test any bit of the first 256 memory locations and jump conditional within an 8 bit PC-relative displacement.

Example: BTJT <ea>, #b, ee. This corresponds to the relative jump (displacement = ee) if bit number #b of memory location <ea> is set. (Bit test and jump if true).

The list of the instructions of this group is given in Table 12.

Group 4 : PC-Relative Jump Group These instructions execute a PC-relative jump (8-bit displacement) depending on the state of the flag bits inside the condition code register (H, I, N, Z, C flags).

Example: JRC ee. This means jump with displacement ee from actual the PC value if the carry bit is set, else execute the next instruction.

The list of the instructions of this group is given in Table 13.

Group 5 : Miscellaneous Group These instructions are mainly control instructions on registers, stack, interrupts, subroutines and power down modes.

The multiply instruction is included in this group. It performs an 8 x 8 bit unsigned multiplication between one index register and the accumulator. The result is given as 16 bits, with the high order byte in the index register and the low order one in the accumulator.

The list of the instructions of this group is given in Table 14.

3.1.2 Addressing Modes

The CPU uses 9 main addressing modes to provide the programmer with an opportunity to optimize his code in most applications.

The various indexed addressing modes make it possible to locate data labels, code conversion tables and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) allow access of tables throughout memory.

Short absolute (direct) and long absolute (extended) addressing modes are also included. Extended addressing permits jump instructions to reach all memory.

The various addressing modes differ from each other in computing the effective address (EA) i.e. in calculating the address to or from which the argument of an instruction is fetched or stored. The LSBEA is the least significant byte of the EA; the MSBEA is its most significant byte. The 17 addressing modes of the processor are described below.

The table of symbols is given in Table 9 while the effective address coding is given in Table 8.

In order to extend the number of op-codes available for an eight bit CPU (256 op-codes), three "pre-byte" op-codes have been defined. These pre-byte have to be seen as pre-instructions that modify the meaning of the instruction they precede. The whole instruction becomes:

PC-1 End of previous instruction

PC Pre-byte

PC + 1 Op-code

PC + 2 Additional word (0 to 2) according to the number of byte required to compute the effective address.

The pre-bytes enable instructions in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or of the instruction using a direct addressing mode.

The pre-bytes are :

PDY : 90h Transform an instruction in X using immediate, direct, indexed, direct bit or inherent addressing modes to an instruction in Y using the same addressing mode.

PIY : 91h Transform an instruction using X indexed addressing mode to an instruction using indirect Y indexed addressing mode.

PIX : 92h Transform an instruction using direct, direct bit or direct relative addressing mode to an instruction using the corresponding indirect addressing mode. It also transforms an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

The pre-byte is completely user transparent. It is part of the assembly code.

Table 8. Source Coding

Addressing Mode		Source Coding	Example
Immediate	#nn	LD a, #0Ah	
Direct	ad8	LD a, 0Ah	
Extended	ad16	LD a, 10EAh	
Indexed no offset	(iX)	LD a, (X)	
Indexed 8 bit offset	(d8, iX)	LD a, (1Bh, Y)	
Indexed 16 bit offset	(d16, iX)	LD a, (100Ah, X)	
Memory indirect short	[ad8]	LD a, [1Bh]	
Memory indirect long	[ad16]	LD a, [100Ah]	
Memory indirect short indexed	([ad8], iX)	LD a, ([1Bh], X)	
Memory indirect long indexed	([ad16], iX)	LD a, ([100Ah], Y)	

Table 9. Table of Symbols used

a	Accumulator	nn	8 bit immediate value
iX	Index register (either X o r Y)	ad8	8 bit address
X	X index register	ad16	16 bit address
Y	Y index register	d8	8 bit signed offset
S	Stack pointer	d16	16 bit signed offset
CC	Condition code register	ee	8 bit PC relative displacement
<ea>	Effective address	b	3 bit number

The addressing modes are discussed in the following paragraphs.

Inherent. In inherent instructions, there is no EA as there is no operand to fetch or store. All the information needed to execute the instruction is contained in the op-code. Operations specifying only an index register or the accumulator, and no other arguments, are included in this mode.

Immediate. In immediate addressing the operand is stored in the byte immediately following the op-code.

Direct Modes

Direct. In the direct addressing mode, the effective address is contained in a single byte following the op-code byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction.

Extended. In the extended addressing mode, the effective address of the argument is contained in the two bytes following the op-code. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory.

Indexed Modes

Indexed Without Offset. In the indexed without offset addressing mode, the effective address is contained in one index register (X or Y). This addressing mode can therefore access the first 256 memory locations. These instructions are only one byte long.

This mode is mainly used to move a pointer through a table or to address a frequently referenced RAM or I/0 location.

Indexed 8 Bit Offset. The EA is obtained by adding the 8-bit unsigned contents of the second instruction byte to the 8-bit unsigned content of the appropriate index register. This mode allows addressing of 256 locations of the 511 lowest memory locations.

Indexed 16 Bit Offset. The EA is obtained by adding the 16-bit unsigned value composed of the second (MSB) and the third (LSB) instruction bytes to the 8-bit unsigned content of the appropriate index register. This mode allows addressing of 256 locations anywhere in the memory map.

Indirect Modes

Short Indirect. In this mode, the second byte of the instruction is used as a page zero address. The content of this page zero memory location is the LSB of the effective address. The MSBEA is zero (i.e. the effective address points to a page zero location).

Long Indirect. In the long indirect mode, the second byte of the instruction is used as a page zero pointer. The MSBEA is the content of this location while the LSBEA is the content of the following page zero location.

Indirect Indexed Modes

Short Indirect Indexed. The second byte of the instruction is used as a page zero address. To obtain the LSBEA, the content of this page zero memory location is added to the 8- bit unsigned value contained in the specified index register. The MSBEA is zero (i.e. the effective address points to a page zero location).

Long Indirect Indexed. In long indirect indexed mode, the second byte of the instruction is used as a page zero pointer. The 16-bit word read from this location (MSB at the pointed address and LSB at the following one) is added to the 8-bit unsigned value contained in the specified index register to form the 16-bit EA.

Relative Modes

Simple Relative. The relative addressing mode is used for branch instruction (e.g. Branch on bit, Branch on condition, Branch subroutine). The branch address (new value of PC) is calculated by adding a displacement given by the 8-bit signed byte following the op-code value to the actual content of the PC. This means that the variation of PC value is in the range -126 to +129 from the op-code address (the offset value can be calculated by the assembler).

Indirect Relative. The indirect relative addressing mode is similar to the relative mode but the content of the second byte of the instruction is used as a page zero address containing the 8-bit signed displacement value to be added to the actual content of the PC (i.e. address of the op-code plus 2).

Direct Bit Modes

Bit Set And Clear. Bit Set/Clear mode is used to read-modify-write one single bit of a memory location in page zero, including I/O bits. The concerned memory location is given by the byte following the op-code (direct addressing mode) while the position of the bit to deal with is given in 3 bits included in the op-code.

Bit Test And Branch. Bit test and branch mode gives a relative branch according to the value of a single bit of a memory location in page zero. The op-code contains 3 bits to define the bit to test at a location given by the byte immediately following the op-code. A third byte gives the 8-bit signed value of the PC-relative displacement. If the test is true, this displacement is added to the actual content of the PC (i.e. op-code address plus 2) to form the new value of the PC.

Indirect Bit Modes

Indirect Bit Set And Clear. Indirect Bit Set/Clear mode works similarly to the bit Set/Clear mode except that the address of the concerned byte is the content of the location pointed by the second byte of the instruction.

Indirect Bit Test And Branch. This mode works as the bit test and branch mode but the tested byte is the content of the location pointed by the second byte of the instruction.

Example

FUNCTION	SOURCE CODING	ADDRESSING MODES
		Immediate
Load a with memory	LD a,<ea >	2 A6 [2]

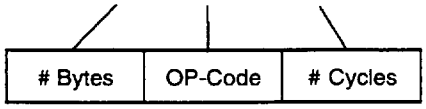


Table 10. Register/Memory and Absolute Jump Group (Group 1)

FUNCTION	SOURCE CODE	ADDRESSING MODE					
		Immediate	Direct	Extended	Index 0	Index 8	Index 16
Load A with memory	LD a,<ea>	2 A6 [2]	2 B6 [3]	3 C6 [4]	1 F6 [3]	2 E6 [4]	3 D6 [5]
Load iX with memory	LD iX,<ea>	2 AE [2]	2 BE [3]	3 CE [4]	1 FE [3]	2 EE [4]	3 DE [5]
Load memory with A	LD <ea>,a	—	2 B7 [4]	3 C7 [5]	1 F7 [4]	2 E7 [5]	3 D7 [6]
Load memory with iX	LD <ea>,iX	—	2 BF [4]	3 CF [5]	1 FF [4]	2 EF [5]	3 DF [6]
Add memory to A	ADD a,<ea>	2 AB [2]	2 BB [3]	3 CB [4]	1 FB [3]	2 EB [4]	3 DB [5]
Add memory and carry to A	ADC a,<ea>	2 A9 [2]	2 B9 [3]	3 C9 [4]	1 F9 [3]	2 E9 [4]	3 D9 [5]
Subtract memory to A	SUB a,<ea>	2 A0 [2]	2 B0 [3]	3 C0 [4]	1 F0 [3]	2 E0 [4]	3 D0 [5]
Subtract memory with carry	SBC a,<ea>	2 A2 [2]	2 B2 [3]	3 C2 [4]	1 F2 [3]	2 E2 [4]	3 D2 [5]
And memory to A	AND a,<ea>	2 A4 [2]	2 B4 [3]	3 C4 [4]	1 F4 [3]	2 E4 [4]	3 D4 [5]
Or memory with A	OR a,<ea>	2 AA [2]	2 BA [3]	3 CA [4]	1 FA [3]	2 EA [4]	3 DA [5]
Exclusive OR	XOR a,<ea>	2 A8 [2]	2 B8 [3]	3 C8 [4]	1 F8 [3]	2 E8 [4]	3 D8 [5]
Arithmetic Compare A	CP a,<ea>	2 A1 [2]	2 B1 [3]	3 C1 [4]	1 F1 [3]	2 E1 [4]	3 D1 [5]
Arithmetic Compare iX	CP iX,<ea>	2 A3 [2]	2 B3 [3]	3 C3 [4]	1 F3 [3]	2 E3 [4]	3 D3 [5]
Bit compare A and memory	BCP a,<ea>	2 A5 [2]	2 B5 [3]	3 C5 [4]	1 F5 [3]	2 E5 [4]	3 D5 [5]
Absolute Jump	JP <ea>	—	2 BC [2]	3 CC [3]	1 FC [2]	2 EC [3]	3 DC [4]
Call subroutine	CALL <ea>	—	2 BD [5]	3 CD [6]	1 FD [5]	2 ED [6]	3 DD [7]

SOFTWARE DESCRIPTION (Continued)

Table 11. Read - Modify - Write Group (Group 2)

FUNCTION	SOURCE CODE	ADDRESSING MODE						
		Inh a	Inh iX	Direct	Memory Direct	Index 0	Index +d8	Index +[ad8]
Increment	INC <ea>	1 4C [3]	1 5C [3]	2 3C [5]	3 923C [7]	1 7C [5]	2 6C [6]	3 926C [8]
(Y index)			2 905C [4]			2 907C [6]	3 906C [7]	3 916C [8]
Decrement	DEC <ea>	1 4A [3]	1 5A [3]	2 3A [5]	3 923A [7]	1 7A [5]	2 6A [6]	3 926A [8]
(Y index)			2 905A [4]			2 907A [6]	3 906A [7]	3 916A [8]
Clear	CLR <ea>	1 4F [3]	1 5F [3]	2 3F [5]	3 923F [7]	1 7F [5]	2 6F [6]	3 926F [8]
(Y index)			2 905F [4]			2 907F [6]	3 906F [7]	3 916F [8]
One's Complement	CPL <ea>	1 43 [3]	1 53 [3]	2 33 [5]	3 9233 [7]	1 73 [5]	2 63 [6]	3 9263 [8]
(Y index)			2 9053 [4]			2 9073 [6]	3 9063 [7]	3 9163 [8]
Negate (2's complement)	NEG <ea>	1 40 [3]	1 50 [3]	2 30 [5]	3 9230 [7]	1 70 [5]	2 60 [6]	3 9260 [8]
(Y index)			2 9050 [4]			2 9070 [6]	3 9060 [7]	3 9160 [8]
Rotate Left thru Carry	RLC <ea>	1 49 [3]	1 59 [3]	2 39 [5]	3 9239 [7]	1 79 [5]	2 69 [6]	3 9269 [8]
(Y index)			2 9059 [4]			2 9079 [6]	3 9069 [7]	3 9169 [8]
Rotate Right thru Carry	RRC <ea>	1 46 [3]	1 56 [3]	2 36 [5]	3 9236 [7]	1 76 [5]	2 66 [6]	3 9266 [8]
(Y index)			2 9056 [4]			2 9076 [6]	3 9066 [7]	3 9166 [8]
Shift Left Logical	SLL <ea>	1 48 [3]	1 58 [3]	2 38 [5]	3 9238 [7]	1 78 [5]	2 68 [6]	3 9268 [8]
(Y index)			2 9058 [4]			2 9078 [6]	3 9068 [7]	3 9168 [8]
Shift Right Logical	SRL <ea>	1 44 [3]	1 54 [3]	2 34 [5]	3 9234 [7]	1 74 [5]	2 64 [6]	3 9264 [8]
(Y index)			2 9054 [4]			2 9074 [6]	3 9064 [7]	3 9164 [8]
Shift Left Arithmetic	SLA <ea>	1 48 [3]	1 58 [3]	2 38 [5]	3 9238 [7]	1 78 [5]	2 68 [6]	3 9268 [8]
(Y index)			2 9058 [4]			2 9078 [6]	3 9068 [7]	3 9168 [8]
Shift Right Arithmetic	SRA <ea>	1 47 [3]	1 57 [3]	2 37 [5]	3 9237 [7]	1 77 [5]	2 67 [6]	3 9267 [8]
(Y index)			2 9057 [4]			2 9077 [6]	3 9067 [7]	3 9167 [8]
Test for Negative or Zero	TNZ <ea>	1 4D [3]	1 5D [3]	2 3D [4]	3 923D [6]	1 7D [4]	2 6D [5]	3 926D [7]
(Y index)			2 905D [4]			2 907D [5]	3 906D [6]	3 916D [7]
Swap Nibbles	SWAP <ea>	1 4E [3]	1 5E [3]	2 3E [5]	3 923E [7]	1 7E [5]	2 6E [6]	3 926E [8]
(Y index)			2 905E [4]			2 907E [6]	3 906E [7]	3 916E [8]

Table 12. Bit Manipulation And Test Group (Group 3)

FUNCTION	SOURCE CODE	ADDRESSING MODES	
		Relative	Indirect Relative
Bit Set	BSET < ea > , # b	2 (10+2*b) [5]	3 92(10+2*b) [7]
Bit Reset	BRES < ea > , # b	2 (11+2*b) [5]	3 92(10+2*b) [7]
Bit Test and Jump if True	BTJT < ea > , # b , ee	3 (00+2*b) [5]	4 92(00+2*b) [7]
Bit Test and Jump if False	BTJF < ea > , # b , ee	3 (01+2*b) [5]	4 92(01+2*b) [7]

Table 13. PC-Relative Jump Group (Group 4)

FUNCTION	SOURCE CODE	ADDRESSING MODES	
		Relative	Indirect Relative
Jump Relative True	JRT ee	2 20 [3]	3 9220 [5]
(Jump Relative always)	JRA ee	2 20 [3]	3 9220 [5]
Jump Relative False	JRF ee	2 21 [3]	3 9221 [5]
Jump Relative if Unsigned Greater than	JRUGT ee	2 22 [3]	3 9222 [5]
Jump Relative if Unsigned Lower or Equal	JRULE ee	2 23 [3]	3 9223 [5]
Jump Relative if No Carry	JRNC ee	2 24 [3]	3 9224 [5]
Jump Relative if Unsigned Greater or Equal	JRUGE ee	2 24 [3]	3 9224 [5]
Jump Relative if Carry	JRC ee	2 25 [3]	3 9225 [5]
Jump Relative if Unsigned Lower than	JRULT ee	2 25 [3]	3 9225 [5]
Jump Relative if Not Equal	JRNE ee	2 26 [3]	3 9226 [5]
Jump Relative if Equal	JREQ ee	2 27 [3]	3 9227 [5]
Jump Relative if Half Carry	JRH ee	2 28 [3]	3 9228 [5]
Jump Relative if Not Half Carry	JRNH ee	2 29 [3]	3 9229 [5]
Jump Relative if Plus	JRPL ee	2 2A [3]	3 922A [5]
Jump Relative if Minus	JRMI ee	2 2B [3]	3 922B [5]
Jump Relative if Not Interrupt Mask	JRNM ee	2 2C [3]	3 922C [5]
Jump Relative if Interrupt Mask	JRM ee	2 2D [3]	3 922D [5]
Jump Relative if Interrupt Line Low	JRIL ee	2 2E [3]	3 922E [5]
Jump Relative if Interrupt Line High	JRIH ee	2 2F [3]	3 922F [5]
Call Subroutine Relative	CALLR ee	2 AD [6]	3 92AD [8]

Table 14. Miscellaneous Group(Group 5)

FUNCTION	SOURCE CODE	X INDEX
Multiply (iX, A = iX * A)	MUL iX , a	1 42 [11]
Load iX with acc. a content	LD iX , a	1 97 [2]
Load a with iX content	LD a , iX	1 9F [2]
Load Stack p. with acc. a content	LD S , a	1 95 [2]
Load acc. a with Stack p. content	LD a , S	1 9E [2]
Load Stack p. with iX content	LD S , iX	1 94 [2]
Load iX with Stack p. content	LD iX , S	1 96 [2]
Load X // with Y // content	LD X , Y	1 93 [2]
Load Y // with X // content	LD Y , X	—
Push acc. a onto the Stack	PUSH A	1 88 [3]
Pop acc. a from the Stack	POP A	1 84 [4]
Push iX onto the stack	PUSH iX	1 89 [3]
Pop iX from the Stack	POP iX	1 85 [4]
Push Condition Codes onto the Stack	PUSH CC	1 8A [3]
Pop Condition Codes from the Stack	POP CC	1 86 [4]
Reset Carry Flag	RCF	1 98 [2]
Set Carry Flag	SCF	1 99 [2]
Reset Interrupt Mask	RIM	1 9A [2]
Set Interrupt Mask	SIM	1 9B [2]
Reset Stack Pointer	RSP	1 9C [2]
No Operation	NOP	1 9D [2]
Interrupt Routine Return	IRET	1 80 [9]
Subroutine Return	RET	1 81 [6]
Software Trap	TRAP	1 83 [10]
Halt	HALT	1 8E [2]
Wait For Interrupt	WFI	1 8F [2]

ST7271 ASSEMBLER REGISTER DEFINITION INCLUDE FILE

```
            BYTES       segment byte at 00-4F 'periph'

.PADR       ds.b        1       ; Port A data register
.PBDR       ds.b        1       ; Port B data register
.PCDR       ds.b        1       ; Port C data register
.PDDR       ds.b        1       ; Port D data register
.PADDR      ds.b        1       ; Port A data direction reg
.PBDDR      ds.b        1       ; Port B data direction reg
.PCDDR      ds.b        1       ; Port C data direction reg
.PDDDR      ds.b        1       ; Port D data direction reg
.ADCDR      ds.b        1       ; ADC data register (port B)
.ADCCR      ds.b        1       ; ADC control register
            ds.b        3       ; reserved
.EWPCC0     ds.b        1       ; East/West control 0
.EWPCC1     ds.b        1       ; East/West control 1
.EEP0CR     ds.b        1       ; E/W EEPROM control reg
.EEP1CR     ds.b        1       ; EEPROM 1 control reg
.EEP2CR     ds.b        1       ; EEPROM 2 control reg
.TIMCR      ds.b        1       ; Timer control register
.TIMSR      ds.b        1       ; Timer status register
.IC1HR      ds.b        1       ; Input capture 1 high reg
.IC1LR      ds.b        1       ; Input capture 2 low reg
.OC1HR      ds.b        1       ; Output compare1 high reg
.OC1LR      ds.b        1       ; Output compare1 low reg
.CNTHR      ds.b        1       ; Counter high register
.CNTLR      ds.b        1       ; Counter low register
.ACNTHR     ds.b        1       ; Alternate cnt high reg
.ACNTLR     ds.b        1       ; Alternate cnt low reg
.IC2HR      ds.b        1       ; Input capture 2 high reg
.IC2LR      ds.b        1       ; Input capture 2 low reg
.OC2HR      ds.b        1       ; Output compare 2 high reg
.OC2LR      ds.b        1       ; Output compare 2 low reg
.PWM0       ds.b        1       ; PWM0 register
.BRM0       ds.b        1       ; BRM0 register (12 bits)
.PWM1       ds.b        1       ; PWM1 register
.BRM1       ds.b        1       ; BRM1 register (12 bits)
.PWM2       ds.b        1       ; PWM2 register
.BRM2_3     ds.b        1       ; BRM2&3 register
.PWM3       ds.b        1       ; BRM3 register
```

```
.PWM4        ds.b        1      ; PWM4 register
.BRM4_5      ds.b        1      ; BRM4&5 register
.PWM5        ds.b        1      ; BRM5 register
.PWM6        ds.b        1      ; PWM6 register
.BRM6_7      ds.b        1      ; BRM6&7 register
.PWM7        ds.b        1      ; BRM7 register
.PWM8        ds.b        1      ; PWM8 register
.BRM8_9      ds.b        1      ; BRM8&9 register
.PWM9        ds.b        1      ; BRM9 register
.PWM10       ds.b        1      ; PWM10 register
.BRM10_11    ds.b        1      ; BRM10&11 register
.PWM11       ds.b        1      ; BRM11 register
.PWM12       ds.b        1      ; PWM12 register
.BRM12_13    ds.b        1      ; BRM12&13 register
.PWM13       ds.b        1      ; BRM13 register
.PWM14       ds.b        1      ; PWM14 register
.BRM14_15    ds.b        1      ; BRM14&15 register
.PWM15       ds.b        1      ; BRM15 register
.PWM16       ds.b        1      ; PWM16 register
.BRM16_17    ds.b        1      ; BRM16&17 register
.PWM17       ds.b        1      ; BRM17 register
.SYNCMCR     ds.b        1      ; Sync processor MCR
.SYNCCCR     ds.b        1      ; Sync processor CCR
.EWDAC       ds.b        1      ; Ewpcc DAC register
             ds.b        1      ; reserved
.SPIDTAIOR   ds.b        1      ; SPI register not used
             ds.b        1      ; reserved
.SPICTRLR    ds.b        1      ; SPI control register
.SPISTSR     ds.b        1      ; SPI status register
.MISCR       ds.b        1      ; Miscelleanous reg
.PBICFGR     ds.b        1      ; Port B input config reg
.PRGIOCFGR   ds.b        1      ; Programmable I/O config

             segment byte at 50-BF 'ram'
             segment byte at C0-FF 'stack'
             WORDS
             segment byte at 100-14F 'ram2'
             segment byte at 150-44F 'eeprom'
             segment byte at 450-3EFF 'rom'
             segment byte at 3FF0-3FFF 'vectit'
```

3.2 ELECTRICAL CHARACTERISTICS

The ST7271 device contains circuitry to protect the inputs againts 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 :

$V_{SS} \leq V_{IN}$ 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 voltage in the following tables are referenced to V_{SS}

ABSOLUTE MAXIMUM RATINGS (Voltage Referenced to Vss)

Symbol	Ratings	Value	Unit	
V_{DD}	Supply Voltage	-0.3 to +7.0	V	
V_{IN}	Input Voltage	V	ss -0.3 to V_{DD} + 0.3	V
V_{DDA}	Analog Reference Voltage	-0.3 to +9.0	V	
I	Current Drain Per Pin Excluding V_{DD} and V_{SS}	25		mA
T_A	Operating Temperature Range Suffix Option 1 (Standard)	T_L to T_H 0 to +70		°C
T_{STG}	Storage Temperature Range	-65 to +150		°C

ELECTRICAL CHARACTERISTICS (Continued)

T_J, the average chip-junction temperature in Celsius can be calculated from the following equation:

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

Where:

- T_A is the Ambient Temperature in °C,
- qJA ithe Package Thermal Resistance, Junction-to-Ambient in °C/W,
- P_D the sum of P_{INT} and $P_{I/O}$,
- P_{INT} equals I_{CC} time V_{CC}, Watts-Chip Internal Power
- $P_{I/O}$ the Power Dissipation on Input and Output Pins, User Determined.

For most applications $P_{I/O} < P_{INT}$ and can be neglected .

P_{PORT} 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:

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

Therefore :

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

Where K is constant pertaining to the particular part, K can be determined from equation (3) by measuring P_D (at equilibrium) for a kwon T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

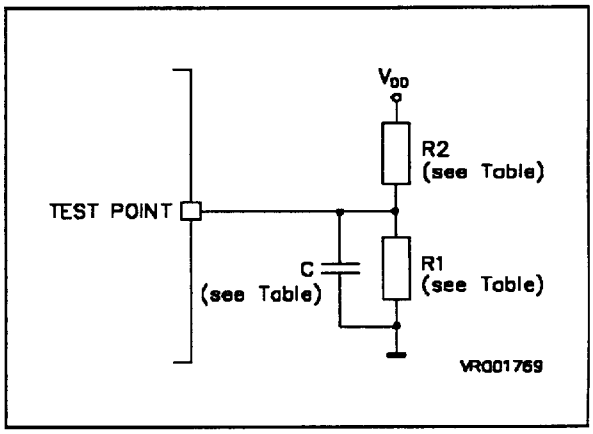
Thermal Characteristics

Symbol	Characteristics	Value	Unit	
θJA	Thermal Resistance PSDIP56 PSDIP42			°C/W

Equivalent Test Load

V_{DD} = 4.5V			
Pins	**R1**	**R2**	**C**
PA0-PA7	3.26kΩ	2.38kΩ	50pF
PB0-PB7	3.26kΩ	2.38kΩ	50pF
PC0-PC5	3.26kΩ	2.38kΩ	50pF
PD0-PD4	3.26kΩ	2.38kΩ	50pF

Test Diagram



ELECTRICAL CHARACTERISTICS (Continued)

AC Electrical Characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{DD}	Operating Supply Voltage	RUN Mode STOP Mode EEPROM Write EEPROM Read	4.5 4.5 4.5 4.5		5.5	V
V_{DDA}	Analog Reference Voltage	7.4	8	8.6	V	
I_{DD}	Supply Current [1]	RUN Mode V_{DD}=5V,f_{OSC}=8MHz		1.8	2.5	mA
		WAIT Mode V_{DD}=5V,f_{OSC}=8MHz		0.9	1.5	mA
		EEPROM Programming V_{DD}=5V,f_{OSC}=8MHz			2.5	µA
		STOP Mode V_{DD}=5V,T_A=70°C		1	10	µA
V_{OL}	Output Low Voltage	Port A, B - Open Drain I_{LOAD} =200µA			0.1xV_{DD}	V

Note 1. Measured with a quartz crystal for the 8 MHz Frequency.

ELECTRICAL CHARACTERISTICS (Continued)

DC Electrical Characteristics

(V_{DD} = 5.0V±10%, V_{SS} = 0V, T_A = operating temperature range, unless otherwise noted)

Symbol	Test Conditions	Min.	Typ.	Max.	Unit
V_{OL}	Output Voltage, Load ≤ 10.0µA	0.1 V			
V_{OH}			V_{DD}-0.1		V
V_{OH}	Output High Voltage				
	I_{LOAD} = 0.8mA, PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4		V_{DD}-0.8		V
V_{OL}	Output Low Voltage				V
	(I_{LOAD} = 1.6 mA) PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$			0.4	
V_{IH}	Input High Voltage				
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$		0.7xV_{DD}		V_{DD} V
V_{IL}	Input Low Voltage				
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$		V_{SS}		0.2xV_{DD} V
V_{RM}	Data Retention Mode (0 to 70°C)	2		V	
I_{IL}	I/O Ports Hi-Z Leakage Current				µA
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4			± 10	
I_{IN}	Input Current : $\overline{RESET}$			± 1	µA
C_{OUT}	Capacitance : Ports (as Input or Output)	12	pF		
C_{IN}	$\overline{RESET}$			8	pF

ELECTRICAL CHARACTERISTICS (Continued)

Control Timing

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Frequency of Operation	V_{DD}=5V	DC		8	MHz
t_{ILCH}	STOP Mode Recovery Startup Time	Ceramic resonator			20	ms
t_{RL}	External RESET Input Pulse Width		1.5		t	CYC
t_{PORL}	Power RESET Ouput	4096		t		CYC
t_{DOGL}	Watchdog RESET Output Pulse Width		1.5		t	CYC
t_{DOG}	Watchdog Time-out	6144		7168	t	CYC
t_{PROG}	EEPROM Programming Time (0 to 70)			2	8	ms
t_{ILIH}	Interrupt Pulse Width PORTB	125	ns			
t_{ILIL}	Interrupt Pulse Period	(1)		t		CYC
t_{DDR}	Power up rise time	V_{DDmin}			100	ms

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

ELECTRICAL CHARACTERISTICS (Continued)

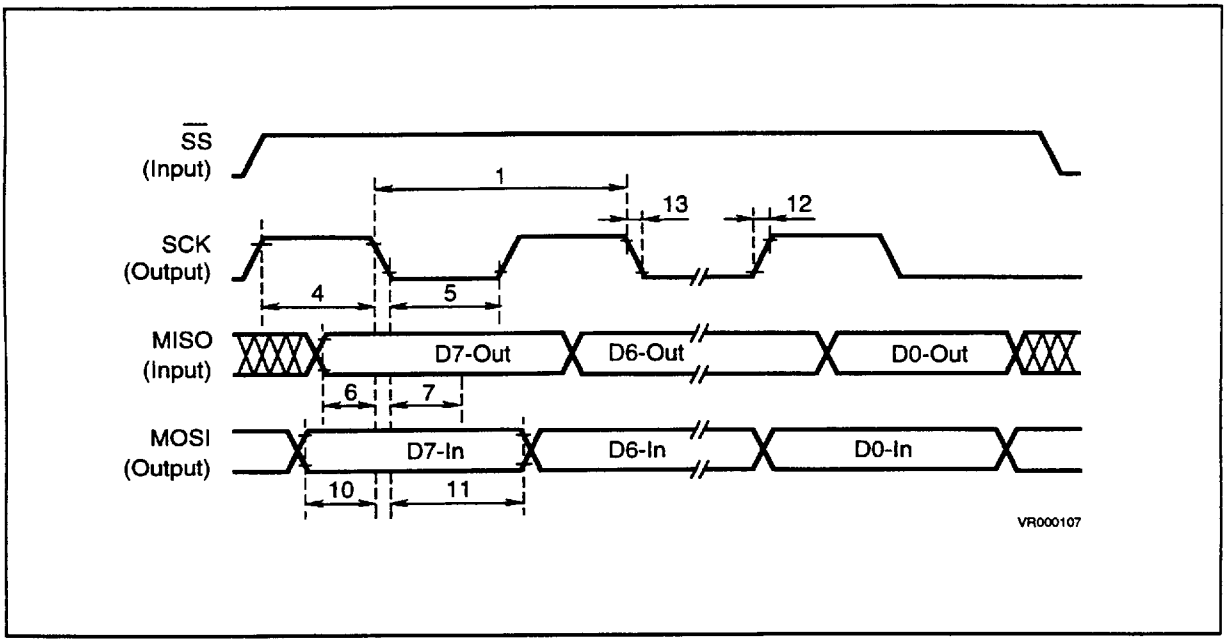
SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 V_{DC} ± 10%, V_{SS} = 0 V_{DC}, T_A = T_L to T_H)

Num.	Symbol	Characteristics	F_{OSC} = 8.0MHz Min.	F_{OSC} = 8.0MHz Max.	Unit
		Operating Frequency = $F_{OSC/2}$ = F_{OP}			
	$f_{OP}(m)$	Master dc 0.5 f			OP
	$f_{OP}(S)$	Slave dc 4.0 MHz			
1		Cycle Time			
	$t_{CYC(m)}$	Master 2.0 tCYC			
	$t_{CYC(s)}$	Slave 240 ns			
2		Enable Lead Time ns			
	$t_{lead(m)}$	Master (1)			
	$t_{lead(s)}$	Slave 120			
3		Enable Lag time ns			
		Master (1)			
		Slave 120			
4		Clock (SCK)High Time			
	$t_{W(SCKH)}$	Master 100 ns			
	$t_{W(SCKH)}$	Slave 90 ns			
5		Clock (SCK) Low Time			
	$t_{W(SCKL)}$	Master 100 ns			
	$t_{W(SCKL)}$	Slave 90 ns			
6		Data Set-up Time			
	$t_{SU(m)}$	Master 100 ns			
	$t_{SU(s)}$	Slave 100 ns			
7		Data Hold Time (Inputs)			
	$t_{H(m)}$	Master 100 ns			
	$t_{H(s)}$	Slave 100 ns			
8	t_A	Access Time (Time to Data Active from High Impedance State)			ns
		Slave 0 120			
9	t_{DIS}	Disable Time (Hold Time to High Impedance State)		ns	
		Slave 240			
10		Data Valid			
	$t_{V(m)}$	Master (Before Capture Edge) 0.25		t	CYC(m)
	$t_{V(s)}$	Slave (After Enable Edge) (2)		120	ns
11		Data Hold Time (Outputs)			
	$t_{HO(m)}$	Master (Before Capture Edge) 0.25		t	CYC(m)
	$t_{HO(s)}$	Slave (After Enable Edge) 0		ns	
12		Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)			
	t_{RM}	SPI Outputs (SCK, MOSI, MISO) 100	ns		
	t_{RS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$) 2.0			µs
13		Fall Time (70% V_{DD} to 20% V_{DD}, C_L			
	t_{FM}	SPI Outputs (SCK, MOSI, MISO) 100	ns		
	t_{FS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$) 2.0			µs

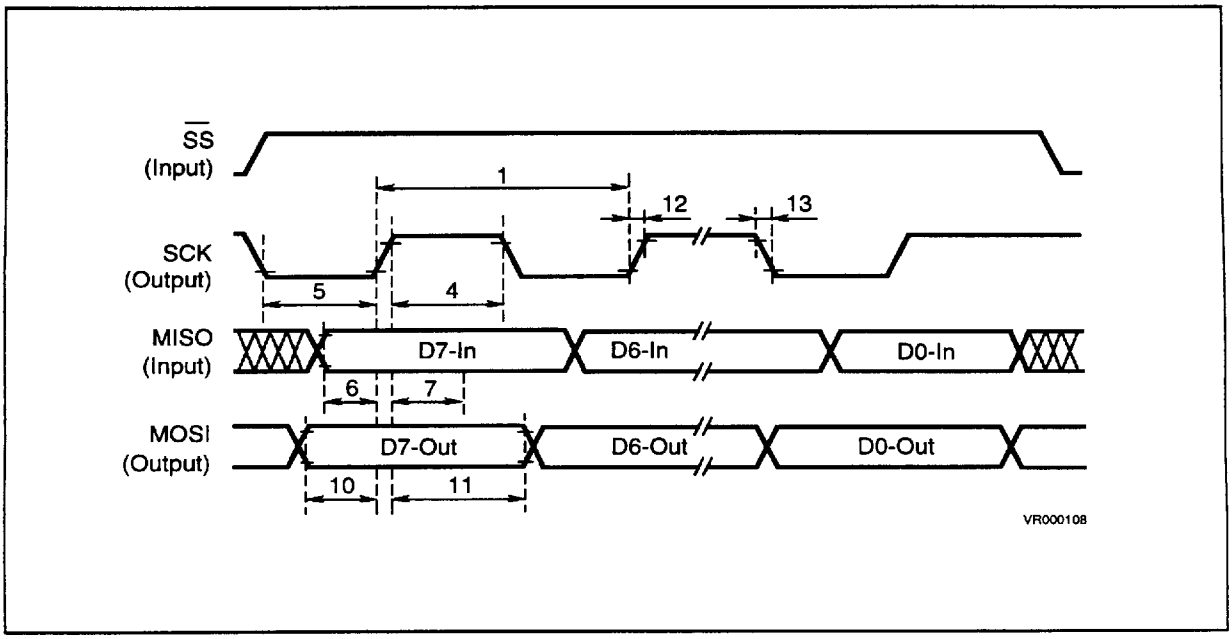
Notes:1 Signal production depends on software. 2 Assumes 200pF load on all SPI pins.

Figure 43. SPI Master Timing Diagram CPOL=0, CPHA=1



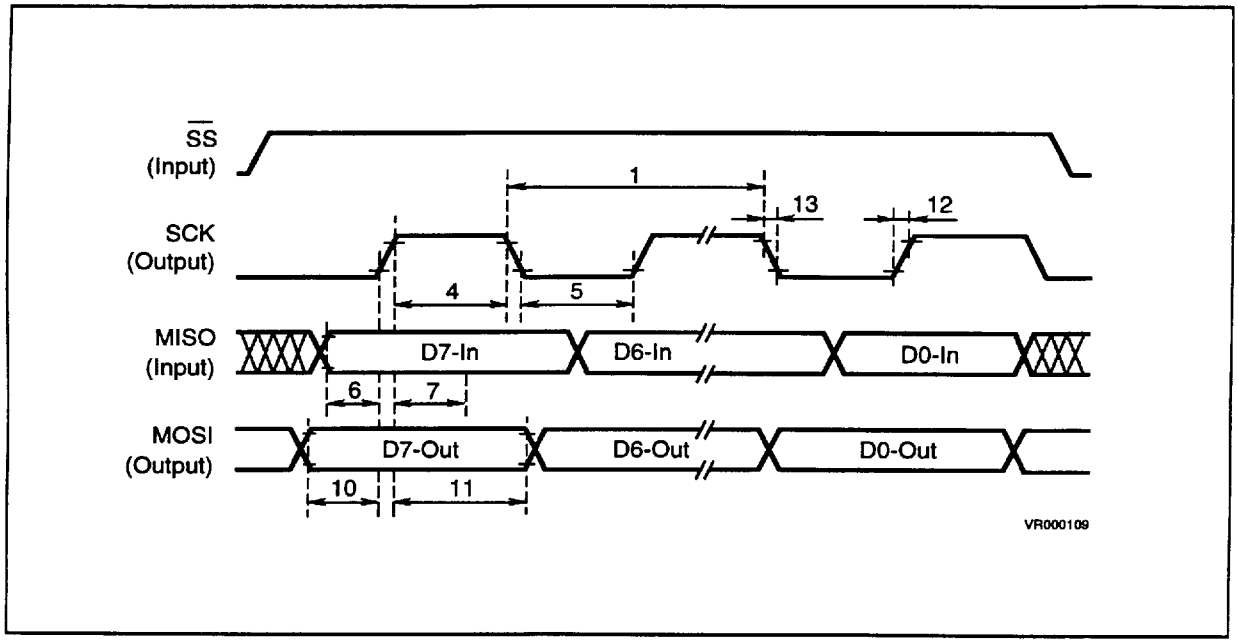
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 44. SPI Master Timing Diagram CPOL=1, CPHA=1



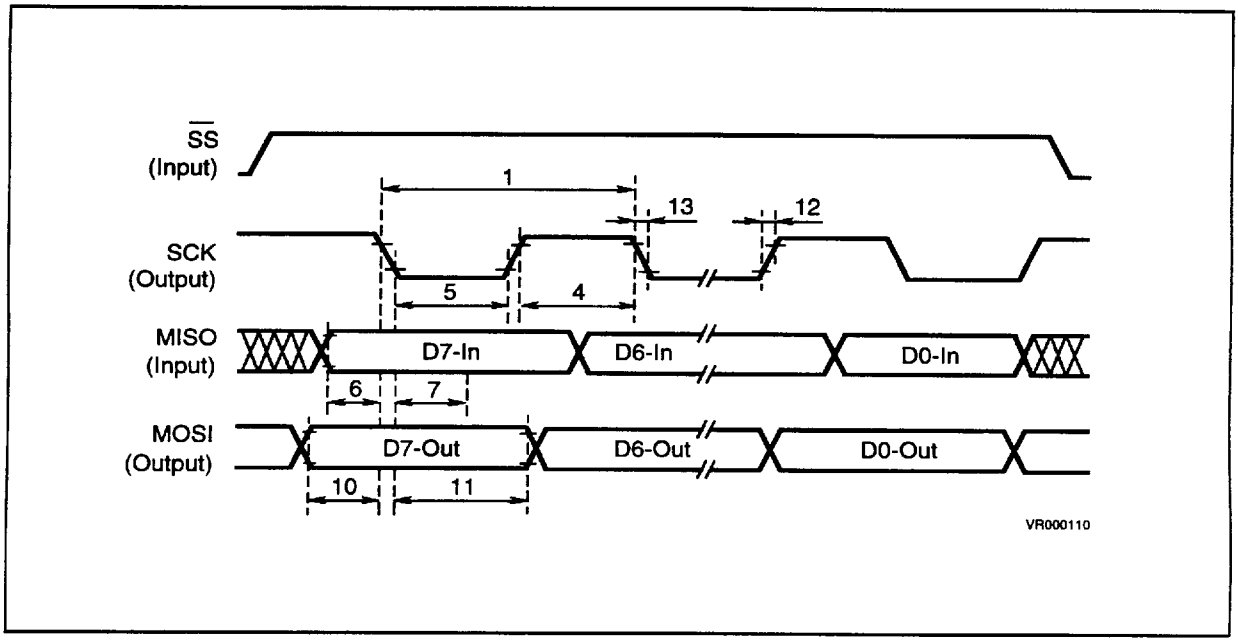
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 45. SPI Master Timing Diagram CPOL=0, CPHA=0



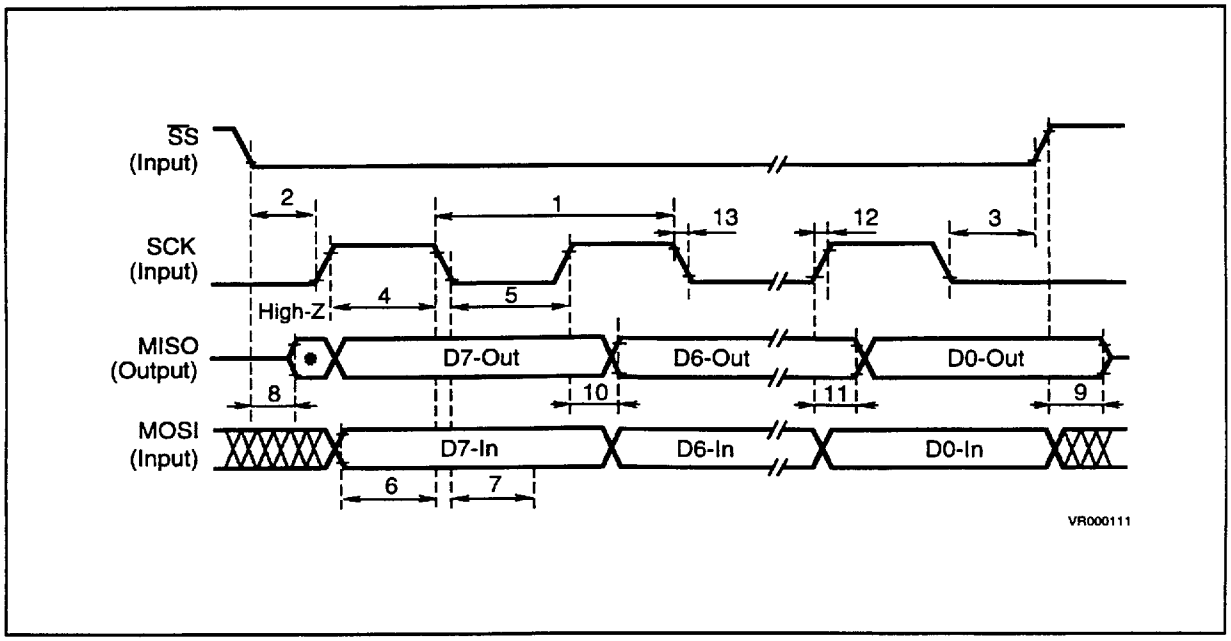
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 46. SPI Master Timing Diagram CPOL=1, CPHA=0



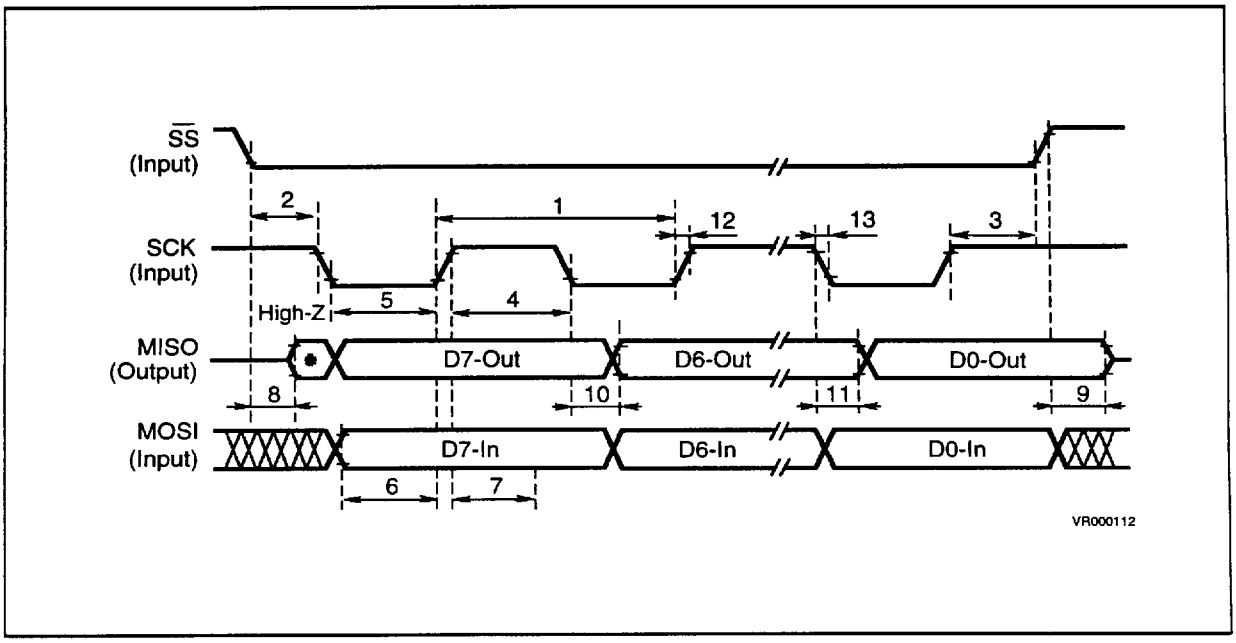
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 47. SPI Slave Timing Diagram CPOL=0, CPHA=1



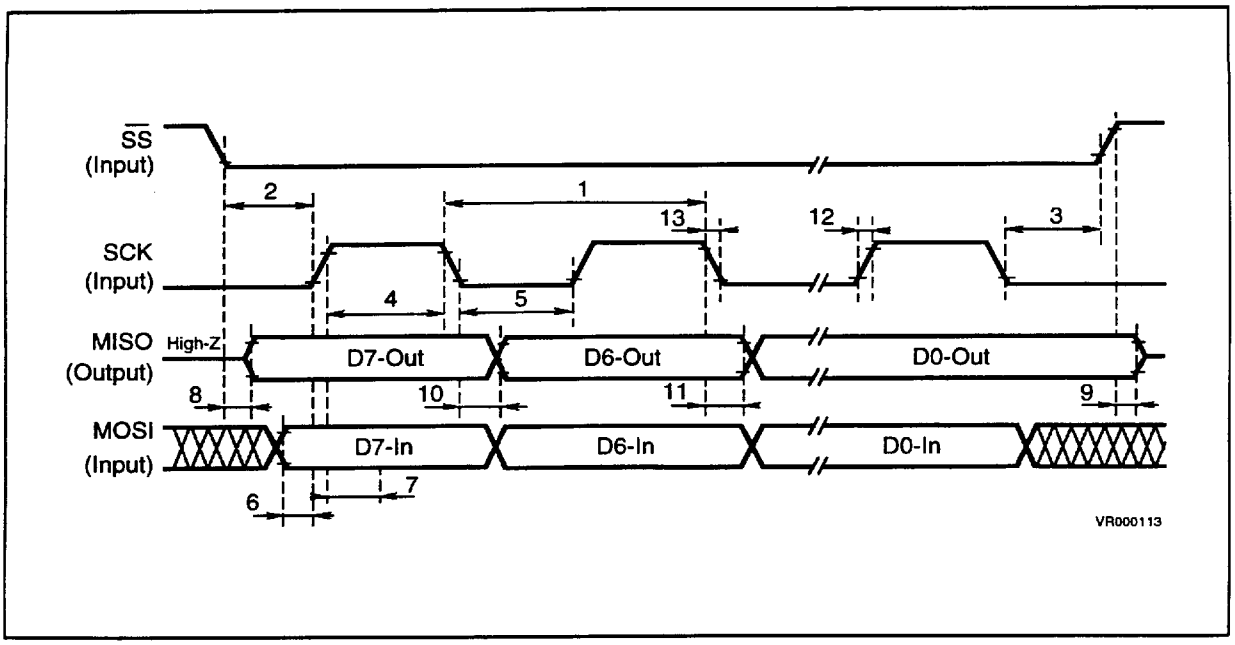
Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

Figure 48. SPI Slave Timing Diagram CPOL=1, CPHA=1



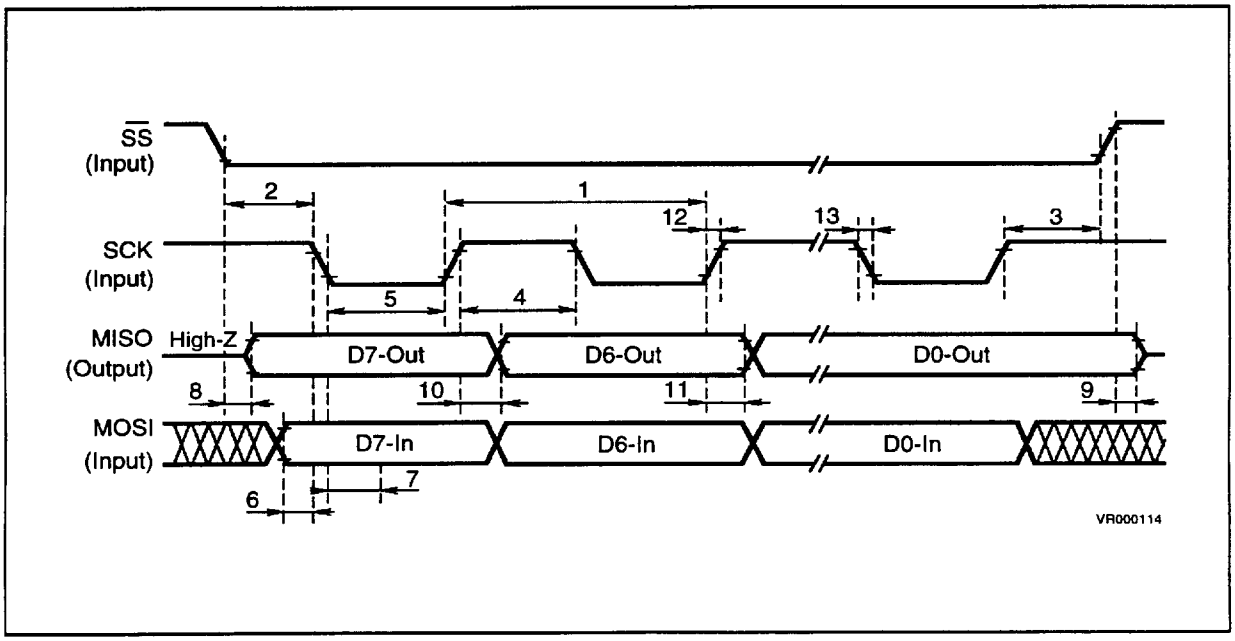
Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

Figure 49. SPI Slave Timing Diagram CPOL=0, CPHA=0



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

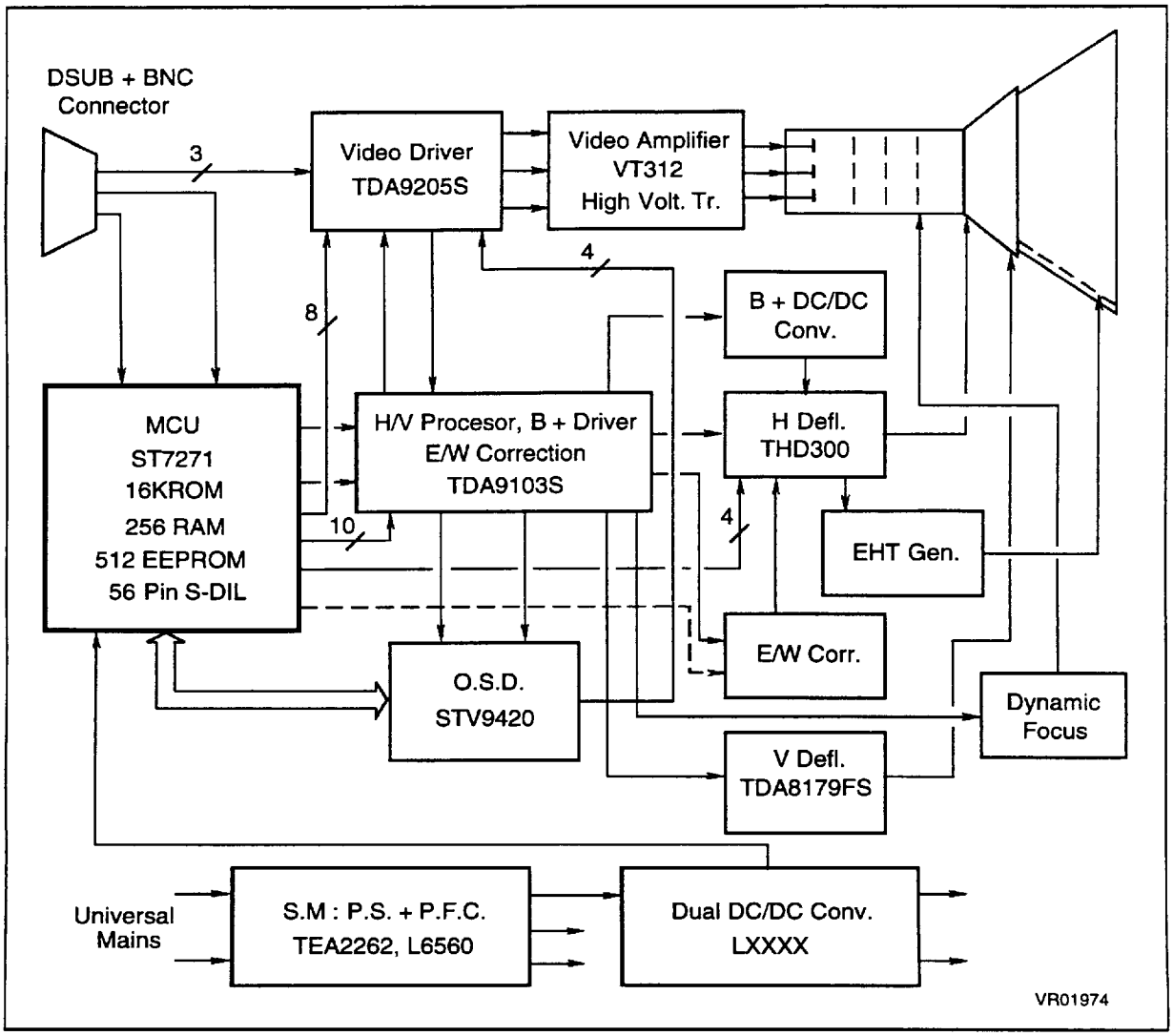
Figure 50. SPI Slave Timing Diagram CPOL=1, CPHA=0



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

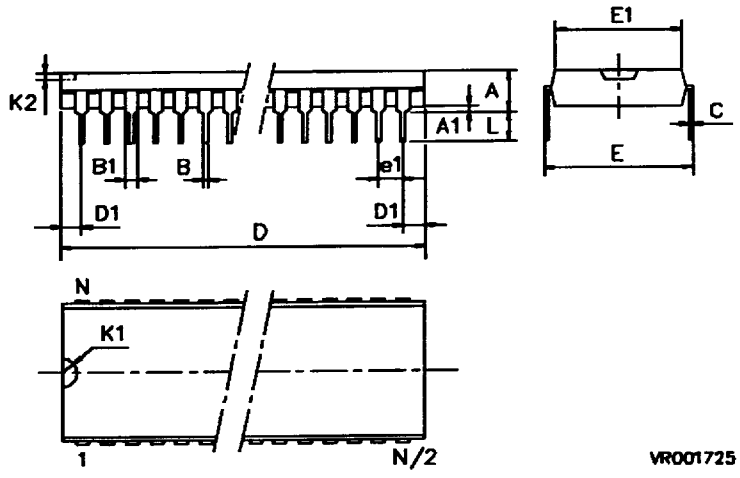
3.3 APPLICATION EXAMPLE

Figure 51. MultiSync 17" to 21" Monitor Block Diagram (with OSD and Dynamic Focus Control)



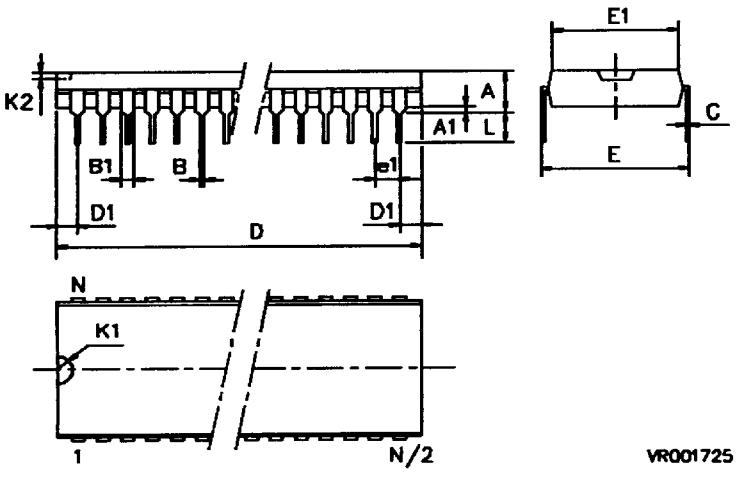
3.4 PACKAGE MECHANICAL DATA

Figure 52. 56-Shrink Plastic Dual In line Package, 600-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D		52.12			2.052	
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	56					

Figure 53. 42-Shrink Plastic Dual In line Package, 600-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

3.5 ORDERING INFORMATION

Ordering Information Table

Sales Type	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	Temperature Range	Package
ST7271N5B1	16K	256	512	-0 to + 70°C	PSDIP56
ST7271N3B1	12K	256	512		PSDIP56
ST7271N1B1	8K	192	384		PSDIP56
ST7271J1B1	8K	192	384		PSDIP42

ST7271 MICROCONTROLLER OPTION LIST

Customer: .....
Address: .....
Contact: .....
Phone No: .....
Reference: .....

SGS-THOMSON Microelectronics references

Device:

PSDIP56	[ ] ST7271N5	[ ] ST7271N3	[ ] ST7271N1
PSDIP42	[ ] ST7271J1		

Temperature Range [ ] 0 to 70°C

Software Development [ ] SGS-THOMSON [ ] Customer
[ ] External Laboratory

For marking one line with 11 characters maximum is possible

Special Marking [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _ _"
Letters, digits, ' .', ' - ', '/ ' and spaces only

OPTION LIST:

Watchdog State After Reset	[ ] Enable	[ ] Disable
Watchdog during WAITmode	[ ] Active	[ ] Suspend
Input Clock to SPI (8MHz osc)	[ ] 2MHz	[ ] 4MHz

Signature .....

Date .....