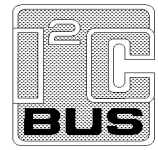
CONTENTS



1 FEATURES

- 80C51 type core
- On-chip oscillator with a maximum frequency of 16 MHz (maximum 0.75 µs instruction cycle)
- A DDC interface:
  - That fully supports DDC1 with specific hardware
  - That is DDC2B, DDC2B+, DDC2AB (ACCESS.bus) compliant, based on a dedicated hardware I²C-bus interface.
  - Contains a specific AUX-RAM buffer with programmable size (128 or 256 bytes) that can be used for DDC operation and shared as system RAM
- Automatic mode detection by hardware to capture the following information:
  - HSYNC frequency with 12-bit resolution
  - VSYNC frequency with 12-bit resolution
  - HSYNC and VSYNC polarity
  - HSYNC and VSYNC presence; needed for the VESA Device Power Management Signalling (DPMS) standard
- On-chip sync processor comprising:
  - Composite sync separation
  - Free running mode
  - Clamping
  - Pattern generation
- Two specific ports for the software I²C-bus interface
- 4 analog voltage outputs derived from an 8-bit Digital-to-Analog Converter (DAC)
- Ten 8-bit Pulse Width Modulation (PWM) outputs for digital control application
- One 14-bit PWM output for digital control application
- One 4-bit Analog-to-Digital Converter (ADC) with 2 input channels (for keyboard interface)
- LED driver port (Port 0); eight port lines with 10 mA drive capability
- One 8-bit port only for I/O function
- 20 derivative I/O ports with the specific port type configuration in each alternative function
- Watchdog Timer with a programmable interval
- On-chip Power-on-reset for low power detection
- Special Idle and Power-down modes for reduced power operation
- Optimized for Electromagnetic Compatibility (EMC)
- Operating temperature: –25 to +85 °C
- Single power supply: 4.4 to 5.5 V.

1.1 Differences from the 80C51 core

- No external memory connection; signals $\overline{EA}$, ALE and $\overline{PSEN}$ are not present.
- Port 1, Port 2 and Port 3 (P3.0 to P3.3 only) mixed with other derivative functions.
- Timer 0/Counter 0 and Timer 1/Counter 1: external input is removed.
- External interrupt 0/INT0 replaced by Mode detection function.
- Standard serial interface (UART) and its control register are removed.
- Wake-up from Power-down mode is also possible by means of an interrupt.

1.2 Memory

Table 1 ROM/RAM sizes

DEVICE	MEMORY	
	ROM	RAM
P83C880	8 kbytes	512 bytes
P83C180	16 kbytes	512 bytes
P83C280	24 kbytes	512 bytes
P83C380	32 kbytes	512 bytes
P87C380 (OTP)	16 kbytes	512 bytes

2 GENERAL DESCRIPTION

The P83Cx80; P87C380 denotes the following types: P83C880, P83C180, P83C280, P83C380 and P87C380, hereafter referred to as the P83C880, are monitor microcontrollers of the 80C51 family, with DDC (DDC1, DDC2B, DDC2B+ and DDC2AB) interface to the PC host. The internal hardware can separate composite sync signals and detect the various display modes. The digital/analog voltage outputs can be used to control the video and deflection functions the monitor.

This data sheet details the specific properties of the P83C880, P83C180, P83C280, P83C380 and P87C380. The shared characteristics of the 80C51 family of microcontrollers are described in *"Data Handbook IC20"*, which should be read in conjunction with this data sheet.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE(1)			TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	VERSION	
P83C880	SDIP42	plastic shrink dual in-line package; 42 leads (600 mil)	SOT270-1	−25 to +85
P83C180				
P83C280				
P83C380				
P87C380 (OTP)				

Note

1. For emulation the package CLCC84 is used.

4 BLOCK DIAGRAM

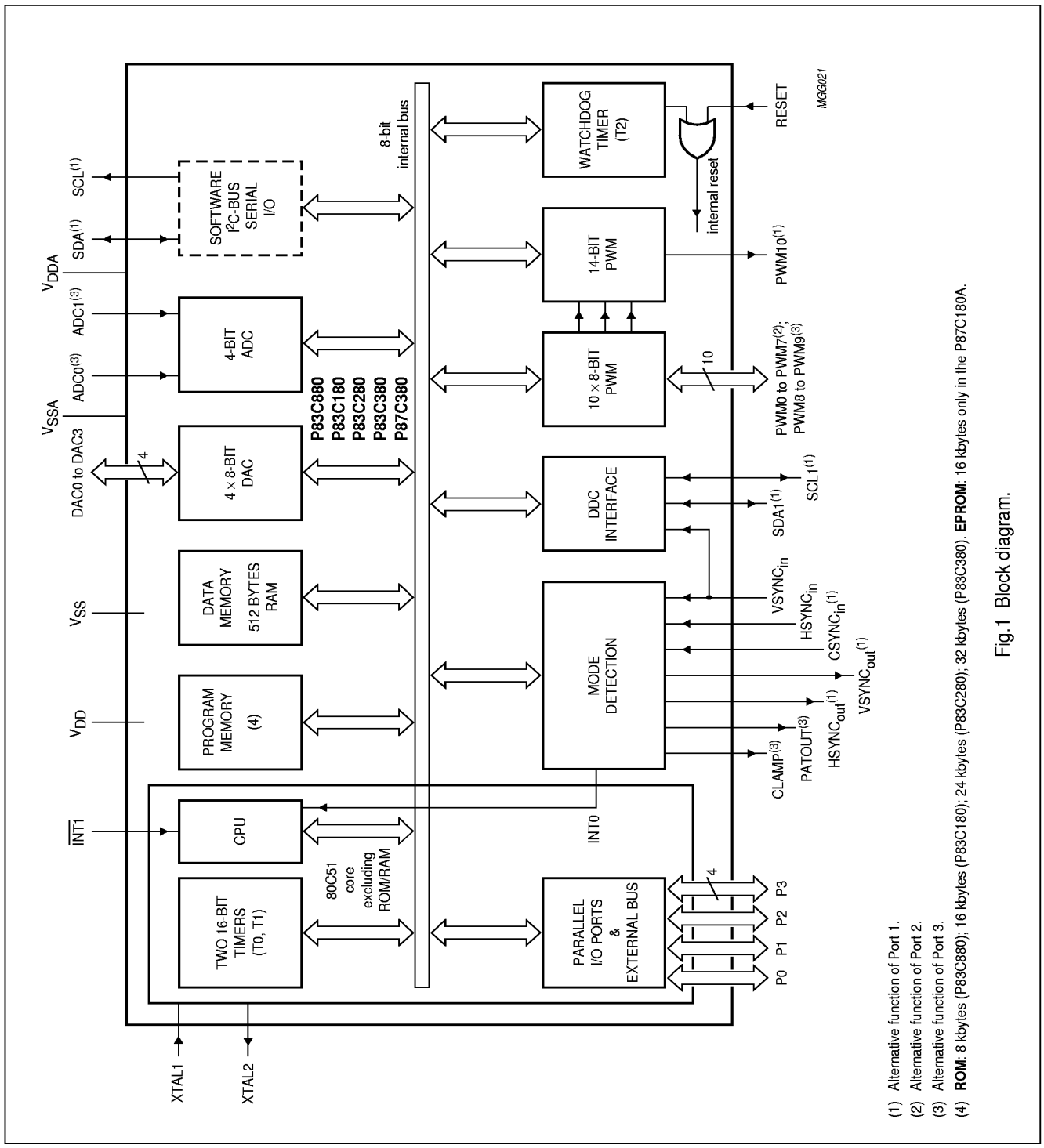


Fig.1 Block diagram.

(1) Alternative function of Port 1.
(2) Alternative function of Port 2.
(3) Alternative function of Port 3.
(4) **ROM**: 8 kbytes (P83C880); 16 kbytes (P83C180); 24 kbytes (P83C280); 32 kbytes (P83C380). **EPROM**: 16 kbytes only in the P87C180A.

5 PINNING INFORMATION

5.1 Pinning

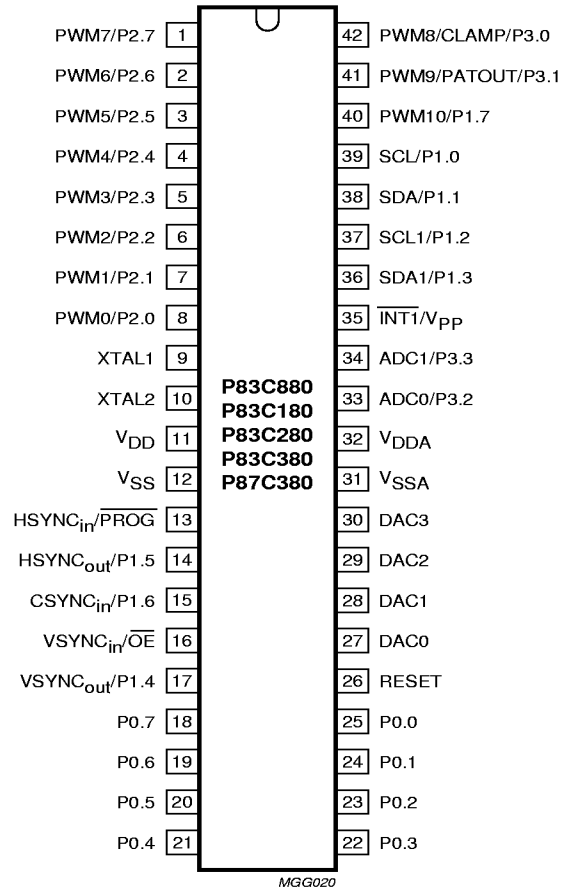


Fig.2 Pinning configuration for SDIP42.

5.2 Pin description

Table 2 Pin description for SDIP42 (SOT270-1)

SYMBOL	PIN	DESCRIPTION
PWM9/PATOUT/P3.1	41	**PWM9 to PWM0**: 8-bit Pulse Width Modulation outputs 9 to 0. Pin 41 and 42 can also be used as the output pin of the test pattern display PATOUT and clamping out signal CLAMP respectively; PATOUT and CLAMP always have the higher priority. Alternative function general I/O ports; **Port 3: P3.1 to P3.0** and **Port 2: P2.7 to P2.0**.
PWM8/CLAMP/P3.0	42	
PWM7/P2.7	1	
PWM6/P2.6	2	
PWM5/P2.5	3	
PWM4/P2.4	4	
PWM3/P2.3	5	
PWM2/P2.2	6	
PWM1/P2.1	7	
PWM0/P2.0	8	
XTAL1	9	Oscillator input pin for system clock.
XTAL2	10	Oscillator output pin for system clock.
V_{DD}	11	Digital power supply (+5 V).
V_{SS}	12	Digital ground.
$HSYNC_{in}/\overline{PROG}$	13	Horizontal sync input pin. During OTP programming it is used as the program pulse input ($\overline{PROG}$).
$HSYNC_{out}$/P1.5	14	Horizontal sync output pin; alternative function: general I/O port P1.5.
$CSYNC_{in}$/P1.6	15	Composite sync input pin; alternative function: general I/O port P1.6.
$VSYNC_{in}/\overline{OE}$	16	Vertical sync input pin. During OTP programming it is used as output strobe ($\overline{OE}$).
$VSYNC_{out}$/P1.4	17	Vertical sync output pin; alternative function: general I/O port P1.4.
P0.7 to P0.0	18 to 25	Port 0: general I/O ports; capability to drive LED.
RESET	26	Reset input; active HIGH initializes the device.
DAC0 to DAC3	27 to 30	8-bit DAC analog voltage output pins; output range: 0 to 5 V.
V_{SSA}	31	Analog ground for DAC and ADC.
V_{DDA}	32	Analog power supply (+5 V) for DAC and ADC.
ADC0/P3.2	33	ADC analog input pins; alternative function: general I/O ports P3.2 and P3.3.
ADC1/P3.3	34	
$\overline{INT1}/V_{PP}$	35	External interrupt input pin. During OTP programming it is used as programming supply voltage pin; V_{PP} = 12.75 V.
SDA1/P1.3	36	I^2C-bus serial data I/O port for the DDC2 interface; alternative function: general I/O port P1.3.
SCL1/P1.2	37	I^2C-bus serial clock I/O port for the DDC2 interface; alternative function: general I/O port P1.2.
SDA/P1.1	38	I^2C-bus serial data I/O port; alternative function: general I/O port P1.1.
SCL/P1.0	39	I^2C-bus serial clock I/O port; alternative function: general I/O port P1.0.
PWM10/P1.7	40	14-bit Pulse Width Modulation output 10; alternative function: general I/O port P1.7.

6 FUNCTIONAL DESCRIPTION

This chapter gives a brief overview of the device. Detailed functional descriptions are given in the following chapters:

Chapter 7 “Memory organization”

Chapter 8 “Interrupts”

Chapter 9 “Watchdog Timer”

Chapter 10 “Input/Output (I/O)”

Chapter 11 “Reduced power modes”

Chapter 12 “Oscillator”

Chapter 13 “Reset”

Chapter 14 “Analog control (DC)”

Chapter 15 “Analog-to-digital converter (ADC)”

Chapter 16 “Digital-to-analog converter (DAC)”

Chapter 17 “Display Data Channel (DDC) interface”

Chapter 18 “I2C-bus Interface”

Chapter 19 “Hardware mode detection”

Chapter 20 “Power management”

Chapter 21 “Control modes”.

6.1 General

The P83C880, P83C180, P83C280, P83C380 and P87C380 8-bit microcontrollers are manufactured in an advanced CMOS process and are derivatives of the 80C51 microcontroller family. They have the same instruction set as the 80C51.

They contain 512 bytes of data memory (RAM). ROM: 8 kbytes (P83C880); 16 kbytes (P83C180); 24 kbytes (P83C280); 32 kbytes (P83C380) and 16 kbytes of EPROM for the P87C180. The microcontrollers are intended for use in monitors ranging from 14" to 21" that can be controlled from the outside (e.g. by a PC) via the external DDC interface.

In addition to the 80C51 standard functions, they provide a number of dedicated hardware functions for monitor application. Eight general I/O ports plus 20 functions combined I/O ports cater for application requirements adequately.

Ten sets of 8-bit PWM deliver the digital waveform for analog control purposes. One 14-bit PWM can support F to V application. The keyboard interface is achieved via a 4-bit ADC. A Watchdog Timer with a maximum count period of 5 s prevents the processor running out of control due to malfunction. Four channels of linear DAC with 8-bit resolution support more accurate analog controls.

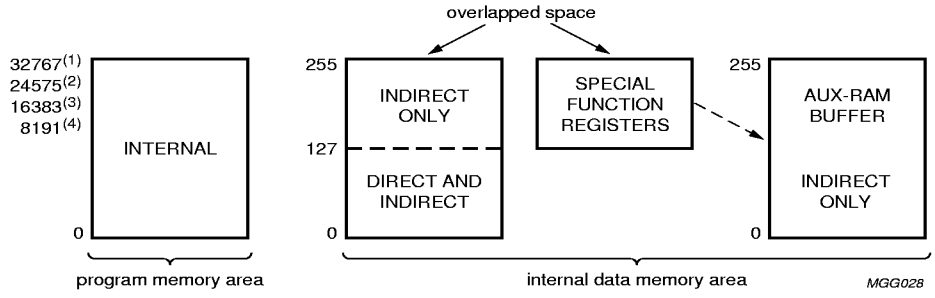
One software I^2C-bus interface is dedicated for the internal connection. A DDC interface will cover all DDC protocols, including DDC1, DDC2B, DDC2AB and DDC2B+.
A hardware mode detector will facilitate mode detection even in power reduced modes, e.g. Idle mode.
The versatile HSYNC and VSYNC outputs can be generated to serve the desired application. In the free running mode, two display patterns can highlight the status of the monitor. Accordingly, the following items will be supported by these microcontrollers:

- Mode detection for:
  - Horizontal sync (HSYNC) frequencies from below 15 kHz up to 150 kHz
  - Vertical sync (VSYNC) frequencies from below 40 Hz up to 200 Hz
- ACCESS.bus interfacing with external devices, e.g. PCs
- DDC1, DDC2B, DDC2AB and DDC2B+ protocols as defined in the VESA DDC standard
- Device Power Management Signalling (DPMS) as described in VESA DPMS proposal.

Figure 1 shows the block diagram functions.

7 MEMORY ORGANIZATION

The Central Processing Unit (CPU) manipulates operands in two memory spaces. There are 512 bytes of internal data memory, consisting of 256 bytes standard RAM and 256 bytes RAM buffer which is accessible as the Auxiliary RAM (AUX-RAM) or addressed through DDCADR and RAMBUF. The memory map and address spaces are shown in Fig.3.



(1) P83C380 and P87C380.
(2) P83C280.
(3) P380C180.
(4) P83C880.

Fig.3 Memory map and address spaces.

7.1 Program memory

The program memory consists of ROM: 8 kbytes (P83C880), 16 kbytes (P83C180), 24 kbytes (P83C280) and 32 kbytes (P83C380). The program memory implemented in the P87C380 is a 16 kbytes EPROM (OTP).

7.2 Internal data memory

The internal data memory is divided into three physically separated parts: 256 bytes of RAM, 256 bytes of AUX-RAM, and a 128 bytes Special Function Registers (SFRs) area. These can be addressed each in a different way as described in Sections 7.2.1 to 7.2.3 and Table 3.

Table 3 Internal data memory map

MEMORY	LOCATION	ADDRESS MODE		POINTERS
		DIRECT	INDIRECT	
RAM	0 to 127(1)	X	X	address pointers are R0 and R1 of the selected register bank
	128 to 255(2)	–	X	
AUX-RAM	0 to 255	–	X(3)	address pointer DDCADR and RAMBUF
SFRs	128 to 255	X	–	–

Notes

1. RAM locations 0 to 127 can be addressed directly and indirectly as in the 80C51.
2. RAM locations 128 to 255 can only be addressed indirectly.

3. Indirect-addressable via MOVX-Datapointer or MOVX-Ri instructions.

7.2.1 RAM

Four register banks, each 8 registers wide, occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack can be located anywhere in the internal 256 bytes RAM. The stack depth is only limited by the available internal RAM space of 256 bytes (see Fig.4).

7.2.2 Special Function Registers (SFRs)

The SFRs can only be addressed directly in the address range from 128 to 255 (see Fig.3). Figure 5 gives an overview of the Special Function Registers space. Sixteen address in the SFRs space are both byte and bit-addressable. The bit-addressable SFRs are those whose address ends in 0H and FH. The bit addresses in this area are 80H to FFH

7.2.3 AUX-RAM

AUX-RAM buffer 0 to 255 is indirectly addressable as external data memory locations 0 to 255 via MOVX-Datapointer instruction or via MOVX-Ri instruction. Since the external access function is not available, any access to AUX-RAM 0 to 255 will not affect the ports.

The 256 bytes of AUX-RAM buffer used to support DDC interface is also available for system usage by indirect addressing through the address pointer DDCADR and data I/O buffer RAMBUF. The address pointer (DDCADR) is equipped with the post increment capability to facilitate the transfer of data in bulk (for details refer to Chapter 17). However, it is also possible to address the AUX-RAM buffer through MOVX command as usually used in the internal RAM extension of 80C51 derivatives.

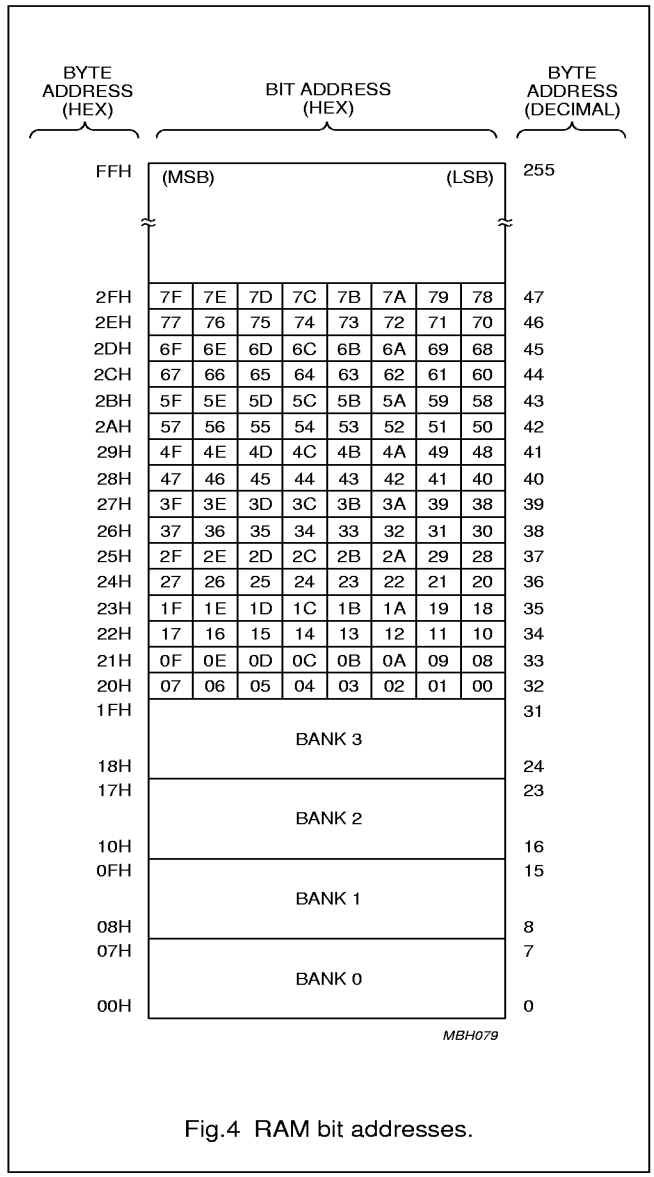


Fig.4 RAM bit addresses.

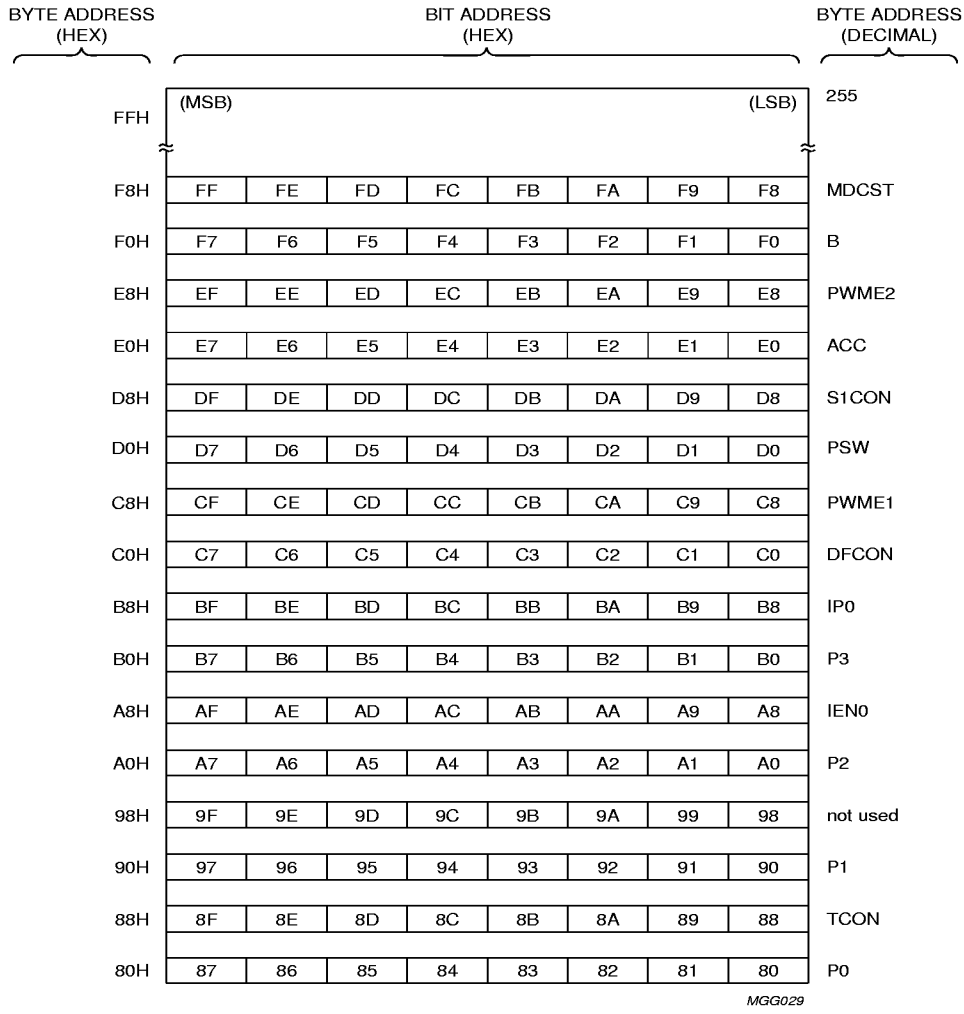


Fig.5 Special Function Registers bit addresses.

7.3 Additional Special Function Registers

The standard SFRs as used in 80C51 and the SFRs for some typical derivative functions like I^2C-bus interface, Timer, etc. are described in the *"Data Handbook IC20"*. The specific SFRs for the P83C880 are introduced in the relevant chapters. Some SFRs which are not mentioned or not dedicated to a certain function will be described in the following sections.

All new additional SFRs used in the P83C880 are listed in Table 4. However, only some of them will be explained in detail in Sections 7.3.1 to 7.3.7.

Table 4 Overview of additional SFRs

REGISTER	DESCRIPTION	ADDRESS	RESET VALUE(1)	READ/WRITE(2)
RAMBUF	RAM Buffer I/O Interface Register	9CH	XXXXXXXX	B
DDCCON	DDC Control Register	9DH	X00X0000	UBBUBBBB
DDCADR	DDC Address Pointer	9EH	00000000	B
DDCDAT	Data Shift Register for DDC1	9FH	00000000	B
DFCON	Miscellaneous Control Register	C0H	10000000	B
ADCDAT	ADC Control Register	C1H	XX000000	UUBBBBBR$_O$
PWM10H	PWM High-byte Data Latch	C6H	00000000	B
PWM10L	PWM Low-byte Data Latch	C7H	10000000	B
S1CON	Control Register for DDC2	D8H	00000000	B
S1STA	Status Register for DDC2	D9H	11111000	R$_o$
S1DAT	Data Shift Register for DDC2	DAH	00000000	B
S1ADR	Address Register for DDC2	DBH	00000000	B
PWME1	PWM Output Control Register 1	C8H	00000000	B
PWME2	PWM Output Control Register 2	E8H	00000000	B
PWM0 to PWM9	Data Latches for 8-bit PWMs	C9H to CFH, EDH to EFH	00000000	B
DAC0 to DAC3	8-bit Data Latches for 8-bit DACs	E9H to ECH	00000000	B
HFP	Free run Control Register for HSYNC$_{out}$	F6H	01100000	B
HFPOPW	Free run and Pulse width for HSYNC$_{out}$	F7H	00011111	B
MDCST	Mode Detect Control and Status Register	F8H	1X000000	BUBBRRRR
VFP	Free run Control Register for VSYNC$_{out}$	F9H	01000000	B
VFPOPW	Free run and Pulse width for VSYNC$_{out}$	FAH	XX000101	B
PULCNT	Pulse Generation Control Register	FBH	00000000	B
HFHIGH	Horizontal Period Counting High-byte Register	FCH	00000000	R$_o$
VFHIGH	Vertical Period Counting High-byte Register	FDH	00000000	R$_o$
VFLHFL	Vertical and Horizontal Period Counting Low-nibbles Register	FEH	00000000	R$_o$
T2	Watchdog Timer Data Register	FFH	00000000	B

Notes

1. X = don't care; even if it's implemented.
2. B = both read/write and R$_o$ = read only; accessible for the entire byte or an individual bit. U = not implemented.

7.3.1 RAM Buffer I/O Interface Register (RAMBUF)

RAMBUF is used as an I/O interface to the RAM buffer. If it is associated with the address pointer DDCADR which is equipped with the capability of post increment, then it will be convenient to transfer the consecutive data stream. This feature is useful to support the DDC/EDID data transfer.

Table 5 RAM Buffer I/O Interface Register (SFR address 9CH)

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

Table 6 Description of RAMBUF bits

BIT	SYMBOL	DESCRIPTION
7 to 0	RAMBUF.7 to RAMBUF.0	8-bit data which is read from or to be written into RAM buffer

7.3.2 Miscellaneous Control Register (DFCON)

This register is bit-addressable.

Table 7 Miscellaneous Control Register (SFR address C0H)

7	6	5	4	3	2	1	0
EW2	SOGE	SYNCE	DDCE	S1E	ADCE	P14LVL	P8LVL

Table 8 Description of DFCON bits

BIT	SYMBOL	DESCRIPTION
7	EW2	**Watchdog Timer enable flag**. This flag is associated with the flags, EW1 (SFR PWM10H) and EW0 in (SFR PWM10L) to form the enable/disable control key for the Watchdog Timer (see Chapter 9). The Watchdog Timer is only disabled by EW2 to EW0 = 101, else it is kept enabled for the rest of the combinations.
6	SOGE	**$CSYNC_{in}$ enable for pin $CSYNC_{in}$/P1.6**. If SOGE = 1, the pin function is $CSYNC_{in}$. If SOGE = 0, the pin function is I/O port P1.6.
5	SYNCE	**Sync separated signals output enable** for pins $VSYNC_{out}$/P1.4 and $HSYNC_{out}$/P1.5. If SYNCE = 1, the pins function as $VSYNC_{out}$ and $HSYNC_{out}$ respectively. If SYNCE = 0, the pins function as I/O ports P1.4 and P1.5 respectively.
4	DDCE	**Enable for DDC interface pins SCL1/P1.2 and SDA1/P1.3**. If DDCE = 1, the pins function as SCL1 and SDA1 respectively for the DDC interface. If DDCE = 0, the pins function as I/O ports P1.2 and P1.3 respectively.
3	S1E	**Enable for I^2C-bus interface pins SCL/P1.0 and SDA/P1.1**. If S1E = 1, the pins function as SCL and SDA respectively for the **I^2C-bus** interface. If S1E = 0, the pins function as I/O ports P1.0 and P1.1 respectively.

BIT	SYMBOL	DESCRIPTION
2	ADCE	**ADC channel enable**. This flag enables the ADC function and also switches the pins ADC0/P3.2 and ADC1/P3.3 to the ADC inputs function. If ADCE = 1, the ADC function is enabled and the pin functions are ADC0 and ADC1 respectively. If ADCE = 0, the ADC function is disabled and the pin functions are I/O ports P3.2 and P3.3 respectively.
1	P14LVL	**Polarity selection bit for the PWM10 output (14-bit PWM)**. If P14LVL = 1, PWM10 output is inverted. If P14LVL = 0, PWM10 output is not inverted.
0	P8LVL	**Polarity selection bit for the PWM0 to PMM9 outputs (8-bit PWM)**. If P8LVL = 1, PWM0 to PWM9 outputs are inverted. If P8LVL = 0, PWM0 to PWM9 outputs are not inverted.

7.3.3 ADC CONTROL REGISTER (ADCDAT)

Table 9 ADC Control Register (SFR address C1H)

7	6	5	4	3	2	1	0
–	–	DACHL	DAC3	DAC2	DAC1	DAC0	COMP

Table 10 Description of ADCDAT bits

BIT	SYMBOL	DESCRIPTION
7 to 6	–	Reserved.
5	DACHL	**ADC input channels selection**. If DACHL = 1, then input channel ADC1 is selected. If DACHL = 0, then input channel ADC0 is selected.
4 to 1	DAC3 to DAC0	**Reference voltage level selection**. The 4 bits select the analog output voltage (V_{ref}) of the 8-bit DAC. For V_{ref} values see Table 31.
0	COMP	**Comparison result; read only**. If COMP = 1, then the ADC input voltage is higher than the reference voltage. If COMP = 0, then the ADC input voltage is lower than the reference voltage.

7.3.4 14-BIT PWM DATA LATCHES (PWM10H AND PWM10L)

Table 11 PWM High-byte Data Latch (PWM10H; SFR address C6H)

7	6	5	4	3	2	1	0
EW1	PWM10H.6	PWM10H.5	PWM10H.4	PWM10H.3	PWM10H.2	PWM10H.1	PWM10H.0

Table 12 Description of PWM10H bits

BIT	SYMBOL	DESCRIPTION
7	EW1	**Watchdog Timer enable flag**. This flag is associated with the flag EW2 (SFR DFCON) and EW0 (SFR PWM10L) to form the enable/disable control key for the Watchdog Timer; see Tables 8 and 14.
6 to 0	PWM10H.6 to PWM10H.0	7 upper data bits for the 14-bit PWM.

Table 13 PWM Low-byte Data Latch (PWM10L; SFR address C7H)

7	6	5	4	3	2	1	0
EW0	PWM10L.6	PWM10L.5	PWM10L.4	PWM10L.3	PWM10L.2	PWM10L.1	PWM10L.0

Table 14 Description of PWM10L bits

BIT	SYMBOL	DESCRIPTION
7	EW0	**Watchdog Timer enable flag**. This flag is associated with the flag EW2 (SFR DFCON) and EW1 (SFR PWM10H) to form the enable/disable control key for the Watchdog Timer; see Table 8 and 12.
6 to 0	PWM10L.6 to PWM10L.0	7 lower data bits for the 14-bit PWM.

7.3.5 PWM Output Control Register 1 (PWME1)

Table 15 PWM Output Control Register 1 (SFR address C8H)

7	6	5	4	3	2	1	0
PWME1.7	PWME1.6	PWME1.5	PWME1.4	PWME1.3	PWME1.2	PWME1.1	PWME1.0

Table 16 Description of PWME1 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	PWME1.7 to PWME1.0	**PWM outputs enable; n = 7 to 0**. If PWME1.n = 1, the corresponding PWM is enabled and pins PWMn/P2.n are switched to PWMn outputs. If PWME1.n = 0, the corresponding PWM is disabled and pins PWMn/P2.n are switched to I/O ports P2.n function.

7.3.6 PWM Output Control Register 2 (PWME2)

Table 17 PWM Output Control Register 2 (SFR address E8H)

7	6	5	4	3	2	1	0
PATENA	DACE3	DACE2	DACE1	DACE0	PWME2.2	PWME2.1	PWME2.0

Table 18 Description of PWME2 bits

BIT	SYMBOL	DESCRIPTION
7	PATENA	**PATOUT (pattern output) enable**. If PATENA = 1, the pin PWM9/PATOUT/P3.1 is switched to the PATOUT (display test pattern) output. If PATENA = 0, the PATOUT function is disabled. The PATOUT function always overrides other alternative functions such as PWM9 and P3.1.
6 to 3	DACE3 to DACE0	**DAC outputs enable (n = 3 to 0)**. If DACEn = 1, the corresponding DACs: DAC3 to DAC0 are enabled. If DACEn = 0, the corresponding DACs: DAC3 to DAC0 are disabled.
2 to 0	PWME2.2 to PWME2.0	**PWM outputs enable; n = 2 to 0**. If PWME2.n = 1, the corresponding PWMs: PWM8, PWM9 and PWM10, are enabled by PWME2.2, PWME2.1 and PWME2.0 respectively and pins PWM8/CLAMP/P3.0, PWM9/PATOUT/P3.1 and PWM10/P1.7 are switched to PWM output. If PWME2.n = 0, the corresponding PWM is disabled. Pins PWM8/CLAMP/P3.0, PWM9/PATOUT/P3.1 and PWM10/P1.7 are switched to I/O port functions P3.0, P3.1 and P1.7 respectively.

7.3.7 Data Latches for 8-bit PWMs (PWM0 to PWM9)

Table 19 Data Latches for 8-bit PWMs (n = 0 to 9; SFR address C9H to CFH and EDH to EFH)

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

Table 20 Description of PWM0 to PWM9 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	PWMn.7 to PWMn.0	8-bit data for PWM channel n (n = 0 to 9)

8 INTERRUPTS

The P83C880 has 5 interrupt sources; these are shown in Fig.6.

Interrupt $\overline{INT1}$ is generated as in a normal 80C51 device. By means of IT1 in SFR TCON this interrupt can be selected to be:

- Level sensitive, when IT1 = LOW; INT1 must be inactive before a return from interrupt instruction (RETI) is given, otherwise the same interrupt will occur again.
- Edge sensitive, when IT1 = HIGH; the internal hardware will reset the latch when the LCALL instruction is executed for the vector address (see Table 21).

Interrupt $\overline{INT0}$ is generated by the mode change of mode detector. Interrupt $\overline{INT0}$ is selected as edge or level sensitive by the state of the IT0 bit in the SFR TCON. However, it is recommended to always set IT0 to HIGH (edge sensitive) so that IE0 will be reset by the internal hardware when the LCALL instruction is executed for the vector address.

Timer 0 and Timer 1 interrupts are generated by TF0 and TF1 which are set by an overflow of their respective Timer/Counter registers (except for Timer 0 in Mode 3; see *"Data Handbook IC20; 80C51 Family; Chapter Timer/Counters"*). When a timer interrupt is generated, the interrupt flag is cleared by the internal hardware when the LCALL instruction is executed for the vector address.

The DDC interrupt is generated either by bit SI (SFR S1CON) for DDC2B/DDC2AB/DDC2B+ protocols or by bit DDC_int (SFR DDCCON) or by bit SWHINT (SFR DDCCON). These flags must be cleared by software.

All bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled in software.

Each of these interrupts sources can be individually enabled or disabled by setting or clearing the bit in Special Function Register IE (see Table 23). IE also contains a global disable bit EA, which disables all interrupts at once.

8.1 Priority level structure

The priority level of each interrupt source can be individually programmed by setting or clearing a bit in Special Function Register IP (see Table 25). A low priority interrupt can itself be interrupted by a high priority interrupt, but not by another low priority interrupt. A high priority interrupt can not be interrupted by another interrupt source.

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If request of the same priority level is received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined as shown in Table 21. The IP register contains a number of reserved (in 80C51) bits: IP.7, IP.6 and IP.4. User software should not write logic 1s to these positions, since they may be used in other 80C51 family products.

Table 21 Priority within levels

SOURCE	PRIORITY WITHIN LEVEL[1]
IE0	1 (highest)
SI	2
TF0	3
IE1	4
TF1	5 (lowest)

Note

1. The 'Priority within level' structure is only used to resolve simultaneous requests of the same priority level.

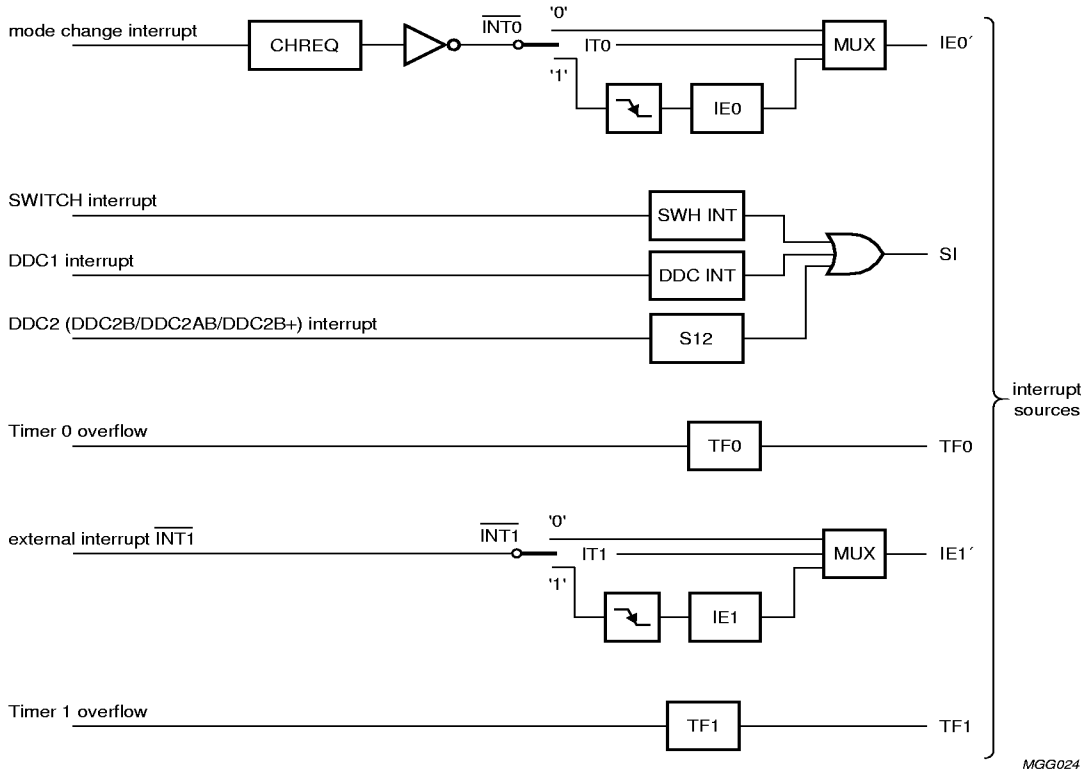


Fig.6 Interrupt sources.

8.2 How interrupts are handled

The interrupt flags are sampled at the S5P2 state of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal priority or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine.

Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before the interrupt is vectored to.The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but not being responded to for one of the above mentioned conditions, and if the flag is still inactive when the blocking condition is removed, then the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in *"Data Handbook IC20; 80C51 family hardware description; Figure: Interrupt Response Timing Diagram"*.

Note that if an interrupt of higher priority level becomes active prior to S5P2 of the machine cycle labelled C3 (see *"Data Handbook IC20; 80C51 family hardware description; Figure: Interrupt Response Timing Diagram"*), then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed. Thus the processor acknowledges an interrupt request by executing a hardware generated LCALL to the appropriate servicing routine. The hardware generated LCALL pushes the contents of the Program Counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 22.

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also return execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress, making future interrupts impossible.

Table 22 Vector addresses

SOURCE	VECTOR ADDRESS
IE0	0003H
SI	002BH
TF0	000BH
IE1	0013H
TF1	001BH

8.3 Interrupt Enable Register (IE)

Table 23 Interrupt Enable Register (SFR address A8H)

7	6	5	4	3	2	1	0
EA	–	ES1	–	ET1	EX1	ET0	EX0

Table 24 Description of IE bits

BIT	SYMBOL	FUNCTION
7	EA	**Disable all interrupts**. If EA = 0, then no interrupt will be acknowledged. If EA = 1, then each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
6	–	Reserved.
5	ES1	**Enable DDC interface interrupt**. If ES1 = 1, then DDC interface interrupt is enabled. If ES1 = 0, then DDC interface interrupt is disabled.
4	–	Reserved.
3	ET1	**Enable Timer 1 overflow interrupt**. If ET1 = 1, then the Timer 1 interrupt is enabled. If ET1 = 0, then the Timer 1 interrupt is disabled.
2	EX1	**Enable external interrupt 1**. If EX1 = 1 then the External 1 interrupt is enabled. If EX1 = 0 then the External 1 interrupt is disabled.
1	ET0	**Enable Timer 0 overflow interrupt**. If ET0 = 1 then the Timer 0 interrupt is enabled. If ET0 = 0 then the Timer 0 interrupt is disabled.
0	EX0	**Enable mode change**. If EX0 = 1 then the mode change interrupt is enabled. If EX0 = 0 then the mode change interrupt is disabled.

8.4 Interrupt Priority Register (IP)

Table 25 Interrupt Priority Register (address B8H)

7	6	5	4	3	2	1	0
–	–	PS1	–	PT1	PX1	PT0	PX0

Table 26 Description of IP bits

BIT	SYMBOL	DESCRIPTION
7 to 6	–	Reserved.
5	PS1	**DDC interface Interrupt priority level**. When PS1 = 1, DDC interface Interrupt is assigned a high priority level.
4	–	Reserved.
3	PT1	**Timer 1 overflow interrupt priority level**. When PT1 = 1, Timer 1 Overflow Interrupt is assigned a high priority level.
2	PX1	**External interrupt 1 priority level**. When PX1 = 1, External Interrupt 1 priority is assigned a high priority level.
1	PT0	**Timer 0 overflow interrupt priority level**. When PT0 = 1, Timer 0 Overflow Interrupt is assigned a high priority level.
0	PX0	**Mode change interrupt priority level**. When PX0 = 1, Mode change Interrupt is assigned a high priority level.

9 WATCHDOG TIMER

In addition to the standard timers, a Watchdog Timer consisting of an 10-bit prescaler and an 8-bit timer is also incorporated. The timer is increased every 19.5 ms for an oscillator frequency of 16 MHz; this is derived from the oscillator frequency (f_{clk}) by the formula:

$$f_{timer} = \frac{f_{clk}}{304 \times 1024}$$

When a timer overflow occurs, the microcontroller is reset. To prevent a system reset, the timer must be reloaded before an overflows occurs, by the application software. If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will produce a reset upon overflow thus preventing the processor running out of control.

The Watchdog Timer can only be reloaded if the condition flag WLE (PCON.4) has been previously set by software. At the moment the counter is loaded the condition flag is automatically cleared.

In the Idle mode the Watchdog Timer and reset circuitry remain active.

The time interval between timer reloading and the occurrence of a reset, depends on the reloaded value.

The Watchdog Timer's time interval is:

$$t = t_1 \times \frac{1024}{(256 - T2)}$$

Where T2 = decimal value of the T2 register contents and t_1 = 15.2 µs (f_{clk} = 10 MHz); t_1 = 12.7 µs (f_{clk} = 12 MHz) and t_1 = 19 µs (f_{clk} = 16 MHz).

For example, this may range from 19.5 ms to 5.0 s when using an oscillator frequency of 16 MHz.

Table 27 lists the resolution and the maximum time interval of the Watchdog Timer using different system clocks.

The Watchdog Timer is controlled by the Watchdog control bits:

- EW2; DFCON.7 (SFR address C0H)
- EW1; PWM10H.7 (SFR address C6H)
- EW0; PWM10L.7 (SFR address C6H).

Only when EW2 to EW0 = 101 the Watchdog Timer is disabled and allows the Power-down mode to be enabled. The rest of pattern combinations will keep the Watchdog Timer enabled and disable the Power-down mode. This security key with multiple flags split in two SFRs will prevent the Watchdog Timer from being terminated abnormally when the function of the Watchdog Timer is needed.

Table 27 Resolution and the maximum time interval of the WDT

f_{clk} (MHz)	PRESCALER FACTOR	RESOLUTION (ms)	MAXIMUM TIME INTERVAL (s)
10	152	15.56	4.0
12		12.97	3.3
16	304	19.46	5.0

10 INPUT/OUTPUT (I/O)

The P83C880 has three 8-bit ports. Ports 0 to 2 are the same as in the 80C51, with the exception of the additional functions of Port 1 and Port 2. Port 3 only contains 4 bits. Port 3 also has alternative functions.

All ports are bidirectional. Pins of which the alternative function is not used may be used as normal bidirectional I/Os.

The use of Port 1, Port 2 and Port 3 pins as an alternative function is carried out automatically by the P83C880 provided the associated Special Function Register bit is set HIGH.

The quasi-bidirectional type of port is applied for Port 1, Port 2 and Port 3. Port 0 is an open-drain I/O port with the capability to drive LED. However, for any port with an alternative function, while the alternative function is performed, the port type will be switched to the appropriate type against a specific function. The port types: quasi-bidirectional, pull-up and open-drain are shown in Figs 7, 8 and 9 respectively.

10.1 The alternative functions for Port 0, Port 1, Port 2 and Port 3

Port 0 Provides the low-order address in programming/verify mode for the P87C380.

Port 1 Used for a number of special functions:

- 2 I/O pins for I^2C-bus interface: SCL/P1.0 and SDA/P1.1. The port type in this situation is set as open-drain.
- 2 I/O pins for DDC interface: SCL1/P1.2 and SDA1/P1.3. The port type in this situation is set as open-drain.
- 2 I/O pins for the outputs of sync separation: $VSYNC_{out}$/P1.4 and $HSYNC_{out}$/P1.5. The port type in this situation is set as push-pull.
- One pin for the composite sync input of sync on green mode: $CSYNC_{in}$/P1.6. There is no pull-up protection diode for this input pin.
- One pin for the 14-bit PWM output: PWM10/P1.7. As PWM function, the port type is open-drain.

Port 2 Two alternative functions are provided:

- High-order address in Programming/Verify mode for P87C380.
- 8 channels of PWM outputs: PWM0/P2.0 to P2.7/PWM7. The port type in this situation is set as open-drain.

Port 3 Two alternative functions are provided:

- Two channels of PWM output: PWM8/CLAMP/P3.0 and PWM9/PATOUT/P3.1. The port type in this situation is set as open-drain. PATOUT and CLAMP functions always override PWM or port function even if they are enabled. For the PATOUT (pattern output) and CLAMP (clamping output) application, the port type is defined as push-pull.
- Two pins for the software ADC input: ADC0/P3.2 and ADC1/P3.3. They are analog inputs.

10.2 EMI (Electromagnetic Interference) reduction

In order to reduce EMI (Electromagnetic Interference) the following design measures have been taken:

- Slope control is implemented on all the I/O lines with alternative functions of the PWM, I^2C-bus and DDC interface. For port pins P1.4 and P1.5, since the alternative functions $VSYNC_{out}$ and $HSYNC_{out}$ are incorporated, the driving capability is made as small as possible to reduce radiation and the slope control function is disabled to have a sharp output. Rise and fall time (10% to 90%) for slope control are:
  $t_{rf(min)}$ < rise/fall time < $t_{rf(max)}$.
  Refer to Chapter 27 for the detailed figures.
- Placing the V_{DD} and V_{SS} pins next to each other
- Double bonding of the V_{DD} and V_{SS} pins, i.e. 2 bondpads for each pin
- Limiting the drive capability of clock drivers and prechargers
- Applying slew rate controlled output drivers
- Internal decoupling of the supply of the CPU core.

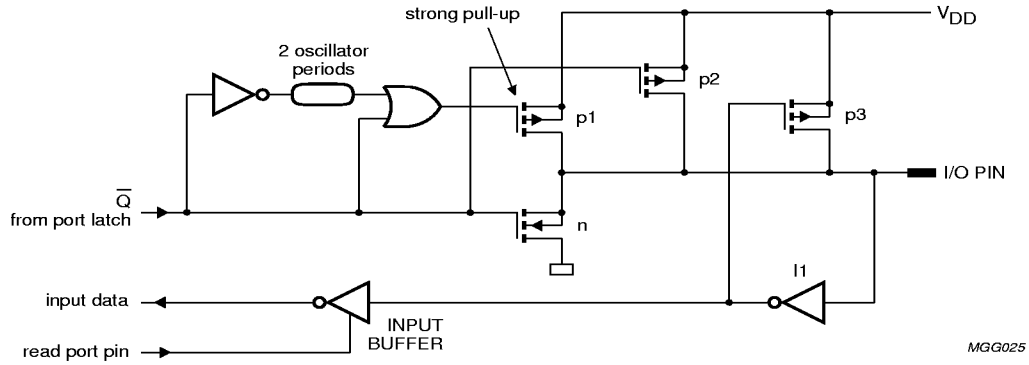


Fig.7 Standard output with quasi-bidirectional port.

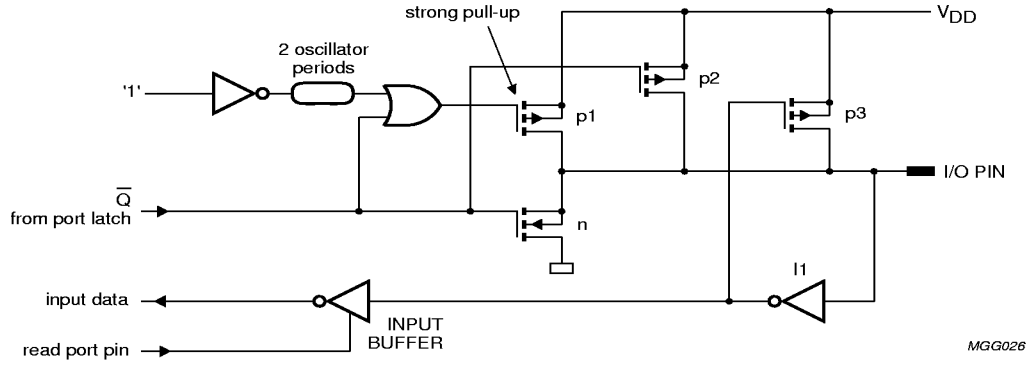


Fig.8 Standard output with the pull-up current source.

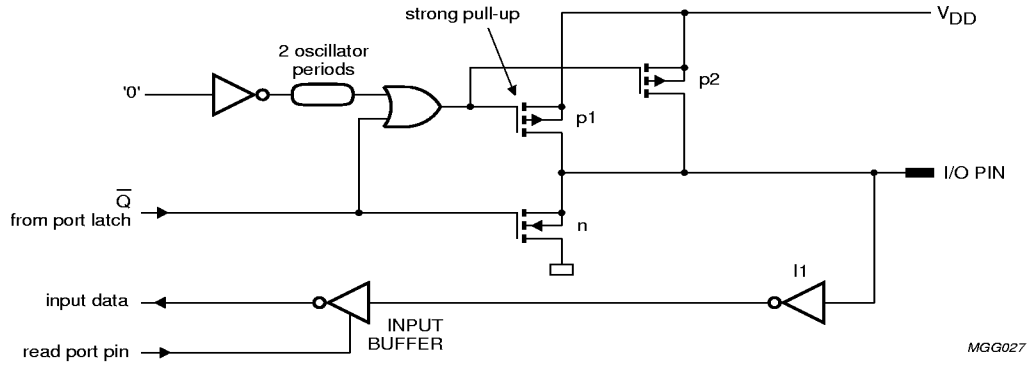


Fig.9 Standard output with the open-drain port.

11 REDUCED POWER MODES

Two software selectable modes of reduced power consumption are implemented. These are the Idle mode and the Power-down mode.

11.1 Power Control Register (PCON)

The Idle mode and Power-down mode are activated by software via the Power Control Register (SFR PCON). Its hardware address is 87H. PCON is not bit addressable. The reset value of PCON is 00H.

11.2 Idle mode

Idle mode operation permits the interrupts, I²C-bus interface, DDC interface, mode detection and timer blocks T0, T1 and T2 (Watchdog Timer) to function while the CPU is halted. The following functions are switched off when the microcontroller enters the Idle mode:

- CPU (halted)
- PWM0 to PWM10 (reset, output = HIGH)
- 4-bit ADC (aborted if conversion is in progress)
- DAC0 to DAC3 (output = indeterminate or frozen at the final value prior to the Idle instruction; decided by software).

The following functions remain active during Idle mode; these functions may generate an interrupt or reset and thus terminate the Idle mode:

- Timer 0, Timer 1 and Timer 2 (Watchdog Timer)
- The DDC interface
- External interrupt
- Mode detection.

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode.

The status of external pins during Idle mode is shown in Table 28.

There are three ways to terminate the Idle mode:

- Activation of any enabled interrupt X0, T0, X1, T1 or S1 will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

  The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during Idle mode. For example, the instruction that writes to PCON.0 can also set or clear both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.
- The second way of terminating the idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.
- The third way of terminating the Idle mode is by an internal watchdog reset.

In all cases the microcontroller restarts after 3 machine cycles.

11.3 Power-down mode

In Power-down mode the system clock is halted. The oscillator is frozen after setting the bit PD in the PCON register.

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the oscillator is stopped.The content of the on-chip RAM and the Special Function Registers are preserved. Note that Power-down mode can not be entered when the Watchdog Timer has been enabled.

The Power-down mode can be terminated by an external reset in the same way as in the 80C51 (but the SFRs are cleared due to RESET) or in addition by the external interrupt, $\overline{\text{INT1}}$.

A termination with $\overline{\text{INT1}}$ does not affect the internal data memory and the Special Function Registers. This gives the possibility to exit from Power-down without changing the port output levels. To terminate the Power-down mode with an external interrupt, $\overline{\text{INT1}}$ must be switched to be level-sensitive and must be enabled. The external interrupt input signal $\overline{\text{INT1}}$ must be kept LOW till the oscillator has restarted and stabilized. The instruction following the one that put the device into the Power-down mode will be the first one which will be executed after the wake-up.

11.4 Status of external pins

- If the $HSYNC_{out}$, $VSYNC_{out}$, PATOUT or CLAMP output is selected (for selection see description in Tables 8 and 18) in Idle or Power-down mode, since sync separation is still alive in Idle mode, $HSYNC_{out}$, $VSYNC_{out}$, PATOUT or CLAMP output will be operating as normal. In Power-down mode: $HSYNC_{out}$, $VSYNC_{out}$, PATOUT or CLAMP output are pulled HIGH.
- In Idle or Power-down mode, if bit DDCE (SFR DFCON) is set, the function of P1.2 and P1.3 will be switched to the DDC interface pins SCL1 and SDA1 respectively. In Idle mode SCL1 and SDA1 can be active only if DDC1 or DDC2 is enabled; otherwise these pins are in the high-impedance (High-Z) state.
- If bit PWME.n (SFR PWME1/PWME2) is set, the function of P1.7, P2.n, P3.0 and P3.1 will be switched to the PWM output function. However, in both Idle and Power-down modes, the output of those PWM pins are pulled HIGH.

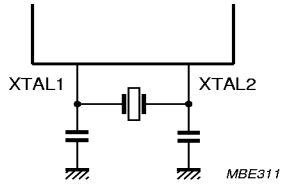
Table 28 Status of external pins during Idle and Power-down modes

MODE	MEMORY	PORT0 TO PORT 3	HSYNC; VSYNC; CLAMP; PATOUT	SCL AND SDA	SCL1 AND SDA1	PWM0 TO PWM10	DAC0 TO DAC3
Idle	internal	data	operative	High-Z	operative	HIGH	unknown
Power-down	internal	data	HIGH	High-Z	High-Z	HIGH	unknown

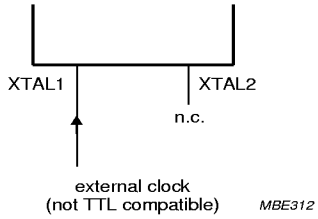
12 OSCILLATOR

The oscillator circuit of the P83C880 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between XTAL1 and XTAL2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuit. Both are operated in parallel resonance.

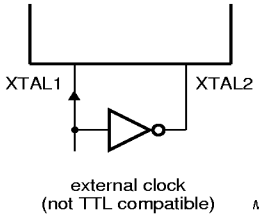
XTAL1 is the high gain amplifier input, and XTAL2 is the output (see Fig.10a). To drive the P83C880 externally, XTAL1 is driven from an external source and XTAL2 left open-circuit (see Fig.10b).



a. Crystal oscillator; C = 20 pF.



b. External clock drive.



c. External clock drive for P87C380.

Fig.10 Oscillator configurations.

13 RESET

There are three ways to invoke a reset and initialize the P83C880:

- Via the external RESET pin
- Via the built-in Power-on-reset circuitry
- Via the Watchdog Timer overflow.

Figure 11 illustrates the reset mechanism. Each reset source will activate an internal reset signal RSTOUT. The CPU responds by executing an internal reset and puts the internal registers in a defined state as shown in Table 29.

Table 29 Reset values of the Special Function Registers

X = Undefined. The internal RAM is not affected by reset.

ADDRESS	REGISTER	CONTENT
80H	P0	1111 1111
81H	SP	0000 0111
82H	DPL	0000 0000
83H	DPH	0000 0000
87H	PCON	0000 0000
88H	TCON	0000 0000
89H	TMOD	0000 0000
8AH	TL0	0000 0000
8BH	TL1	0000 0000
8CH	TH0	0000 0000
8DH	TH1	0000 0000
90H	P1	1111 1111
9CH	RAMBUF	0000 0000
9DH	DDCCON	X00X 0000
9EH	DDCADR	0000 0000
9FH	DDCDAT	0000 0000
A0H	P2	1111 1111
A8H	IEN0	0X0X 0000
B0H	P3	XXXX 1111
B8H	IP0	XX0X 0000
C0H	DFCON	1000 0000
C1H	ADCDAT	XX00 0000
C6H	PWM10H	0000 0000
C7H	PWM10L	1000 0000
C8H	PWME1	0000 0000
C9H	PWM0	0000 0000
CAH	PWM1	0000 0000
CBH	PWM2	0000 0000
CCH	PWM3	0000 0000

ADDRESS	REGISTER	CONTENT
CDH	PWM4	0000 0000
CEH	PWM5	0000 0000
CFH	PWM6	0000 0000
D0H	PSW	0000 0000
D8H	S1CON	0000 0000
D9H	S1STA	1111 000
DAH	S1DAT	0000 0000
DBH	S1ADR	0000 0000
E0H	ACC	0000 0000
E8H	PWME2	0000 0000
E9H	DAC0	0000 0000
EAH	DAC1	0000 0000
EBH	DAC2	0000 0000
ECH	DAC3	0000 0000
EDH	PWM7	0000 0000
EEH	PWM8	0000 0000
EFH	PWM9	0000 0000
F0H	B	0000 0000
F6H	HFP	0110 0000
F7H	HFPOPW	0000 0101
F8H	MDCST	1X00 0000
F9H	VFP	0100 0000
FAH	VFPOPW	XX00 0101
FBH	PULCNT	0000 0000
FCH	HFHIGH	0000 0000
FDH	VFHIGH	0000 0000
FEH	VFLHFL	0000 0000
FFH	T2	0000 0000

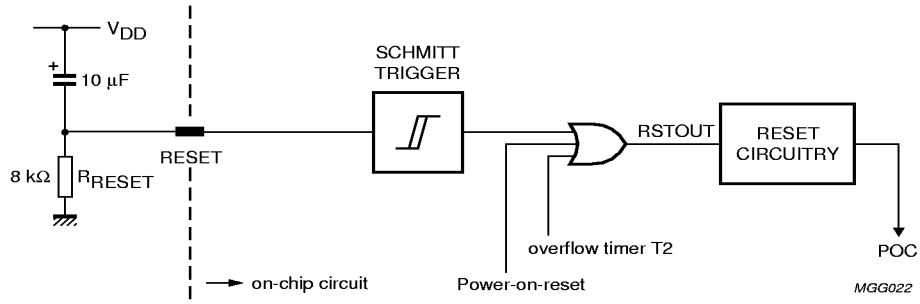


Fig.11 On-chip reset configuration.

13.1 External reset

Pin RESET is connected to a Schmitt trigger for noise reduction (see Fig.11). A reset is accomplished by holding the RESET pin HIGH for at least 16 machine cycles (192 system clocks) while the oscillator is running.

An automatic reset can be obtained by switching on V_{DD}, if the RESET pin is connected to V_{DD} via a capacitor and a resistor as illustrated in Fig.11. The V_{DD} rise time must not exceed 10 ms and the capacitor should be at least 10 μF. The decrease of the RESET pin voltage depends on the capacitor and the external resistor R_{RESET}. The voltage must remain above the lower threshold for at least the oscillator start-up time plus 2 machine cycles.

13.2 Power-on-reset

An on-chip Power-on-reset circuit that detects the supply voltage rise or fall and accordingly generates a power-on reset pulse (see Fig.12).

In the case of supply voltage rise, the power-on reset signal will follow the supply voltage rise; after reaching the trip level V_t the power-on reset signal will maintain the same behaviour, and returns to a LOW state after a time interval T_p.

In the case of supply voltage fall, after the trip level V_t is reached, the power-on reset signal will follow the waveform of the supply voltage for time interval T_p.

The time interval T_p guarantees that a complete power-on reset pulse can trigger the internal reset signal. However, to ensure that the oscillator is stable before the controller starts, the clock signals are gated away from the CPU for a further 2048 oscillator cycles.

The values of V_t, T_p and a process tolerance of $\pm \Delta V_t$ can be found in Chapter 25; currently V_t = 3.9 V, T_p = 10 μs and ΔV_t = 0.3 V.

13.3 T2 (Watchdog Timer Data Register) overflow

The length of the output pulse from T2 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

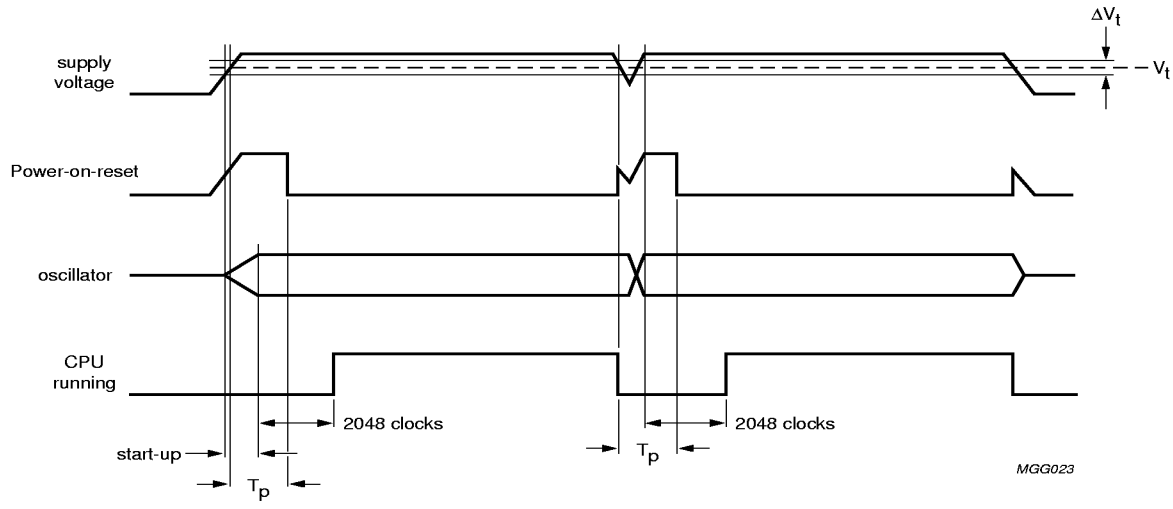


Fig.12 Power-on reset switching level.

14 ANALOG CONTROL (DC)

The P83C880 has eleven Pulse Width Modulated (PWM) outputs for analog control purposes e.g. brightness, contrast, E-W, R (or G or B) gain control etc. Each PWM output generates a pulse pattern with a programmable duty cycle.

The eleven PWM outputs comprise:

- 10 PWM outputs with 8-bit resolution (PWM0 to PWM9); described in Section 14.1.
- 1 PWM output with 14-bit resolution (PWM10); described in Section 14.2.

A typical PWM output application is described in Section 14.3.

14.1 8-bit PWM outputs (PWM0 to PWM9)

PWM outputs PWM0 to PWM9 share the same pins as port lines P2.0 to P2.7, P3.0 and P3.1 respectively. Selection of the pin function as either a PWM output or a port line is achieved using the appropriate PWMnE bit in SFRs, PWME1 (address C8H) and PWME2 (address E8H); see Table 4.

The polarity of the PWM outputs is programmable and is selected by the P8LVL bit in SFR DFCON (address C0H); see Table 4.

The duty cycle of outputs PWM0 to PWM9 is dependent on the programmable contents of the data latches: SFRs PWM0 to PWM9. As the clock frequency of each PWM circuit is $\frac{1}{4}f_{clk}$, the pulse width of the pulse generated can be calculated as: $\text{Pulse width} = \frac{4 \times (\text{PWMn})}{f_{clk}}$

Where (PWMn) is the decimal value held in the relevant data latch.

The maximum repetition frequency of the 8-bit PWM outputs is: $f_{PWM} = \frac{f_{clk}}{1024}$

The block diagram for the 8-bit PWM outputs is shown in Fig.13.

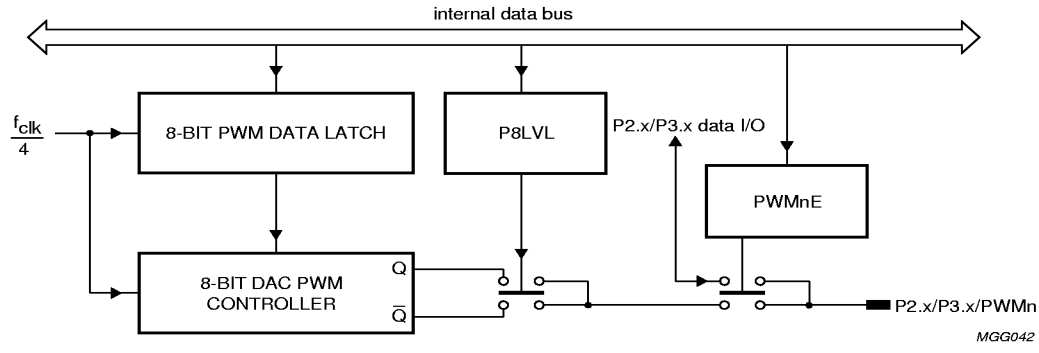


Fig.13 Block diagram for 8-bit PWMs.

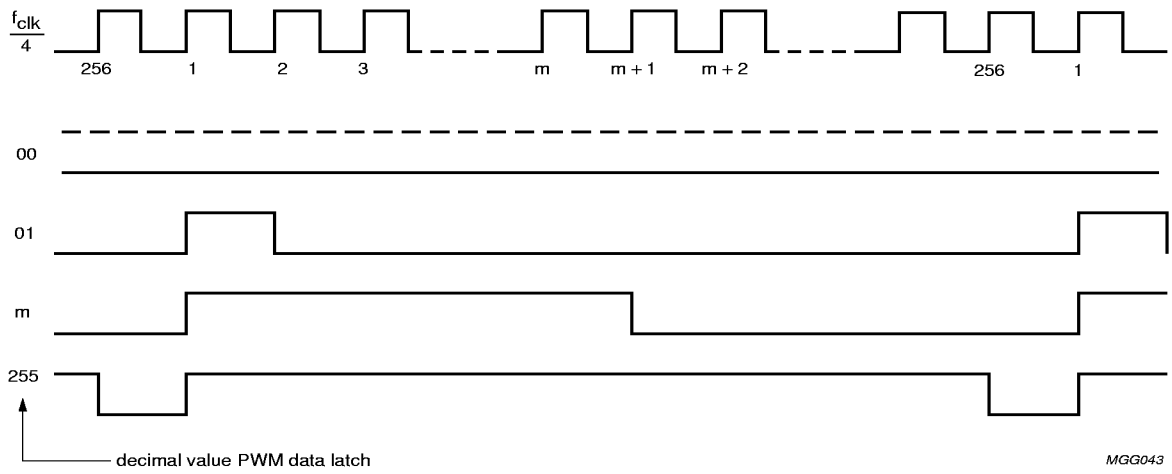


Fig.14 PWM0 to PWM9 output patterns.

14.2 14-bit PWM output (PWM10)

PWM10 shares the same pin as port line P1.7. Selection of the pin function as either a PWM output or as a port line is achieved using the PWME2.0 bit in SFR PWME2 (address E8H); see Table 4.

The block diagram for the 14-bit PWM output is shown in Fig.15 and comprises:

- Two 7-bit latches; SFRs PWM10H and PWM10L
- 14-bit data latch (PWMREG)
- 14-bit counter
- Coarse pulse controller
- Fine pulse controller
- Mixer.

Data is loaded into the 14-bit data latch (PWMREG) from the two 7-bit data latches (PWM10H and PWM10L) when PWM10H is written to. The upper seven bits of PWMREG are used by the coarse pulse controller and determine the coarse pulse width; the lower seven bits are used by the fine pulse controller and determine in which subperiods fine pulses will be added.

The outputs OUT1 and OUT2 of the coarse and fine pulse controllers are then 'ORed' in the mixer to give the PWM10 output. The polarity of the PWM10 output is programmable and is selected by the P14LVL bit in SFR DFCON (address C0H); see Section 7.3.2.

As the 14-bit counter is clocked by $\frac{1}{4}f_{clk}$, the repetition times of the coarse and fine pulse controllers may be calculated as shown below.

Coarse controller repetition time: $t_{sub} = 128 \times \frac{4}{f_{clk}}$

Fine controller repetition time: $t_r = 128 \times 128 \times \frac{4}{f_{clk}}$

Figure 16 shows typical PWM10 outputs, with coarse adjustment only, for different values held in PWM10H, when P14LVL = 1. Figure 17 shows typical PWM10 outputs when P14LVL = 1, with coarse and fine adjustment, after the coarse and fine pulse controller outputs have been 'ORed' by the mixer.

When P14LVL = 1, the PWM10 output is inverted.

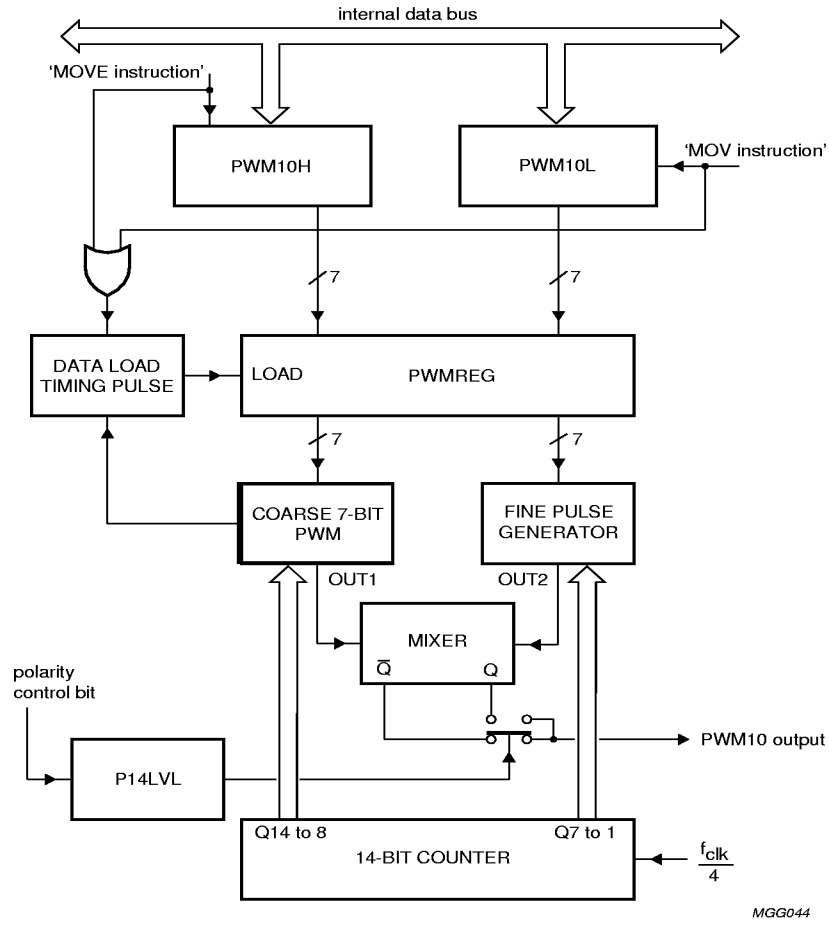


Fig.15 14-bit PWM block diagram.

14.2.1 COARSE ADJUSTMENT

An active HIGH pulse is generated in every subperiod; the pulse width being determined by the contents of PWM10H. The coarse output (OUT1) is HIGH at the start of each subperiod and will remain HIGH until the time $[4/f_{clk} \times (128 - PWM10H)]$ has elapsed. The output will then go LOW and remain LOW until the start of the next subperiod. The coarse pulse width may be calculated as:

$$\text{Pulse duration} = (PWM10H) \times \frac{4}{f_{clk}}$$

14.2.2 FINE ADJUSTMENT

Fine adjustment is achieved by generating an additional pulse in specific subperiods. The pulse is added at the start of the selected subperiod and has a pulse width of $4/f_{clk}$. The contents of PWM10L determine in which subperiods a fine pulse will be added. It is the logic 0 state of the value held in PWM10L that actually selects the subperiods. When more than one bit is a logic 1 then the subperiods selected will be a combination of those subperiods specified in Table 30.

For example, if PWM10L = 000 0101 then this is a combination of:

- PWM10L = 000 0001: subperiod 64
- PWM10L = 000 0100: subperiods 16, 48, 80 and 112.

Pulses will be added in subperiods 16, 48, 64, 80 and 112. This example is illustrated in Fig.18.

When PWM10L holds 000 0000 fine adjustment is inhibited and the PWM10 output is determined only by the contents of PWM10H.

Table 30 Additional pulse distribution

PWM10L	ADDITIONAL PULSE IN SUBPERIOD
000 0001	64
000 0010	32 and 96
000 0100	16, 48, 80 and 112
000 1000	8, 24, 40, 56, 72, 88, 104 and 120
001 0000	4, 12, 20, 28, 36, 44, 52...116 and 124
010 0000	2, 6, 10, 14, 18, 22, 26, 30...122 and 126
100 0000	1, 3, 5, 7, 9, 11, 13, 15, 17...125 and 127

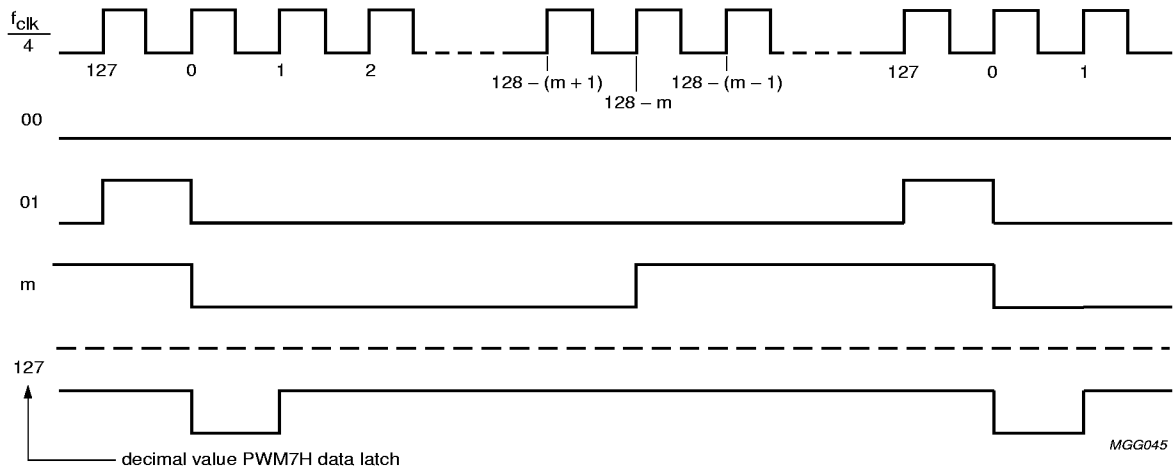


Fig.16 PWM10 output patterns: Coarse adjustment only.

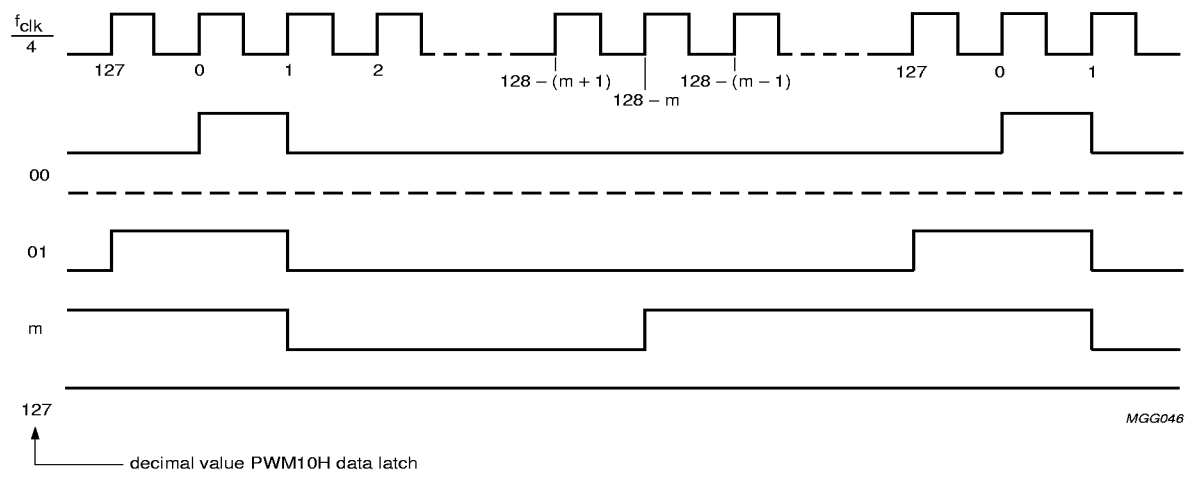


Fig.17 PWM10 output patterns: Coarse and fine adjustment (PWM10L = 000 0000B).

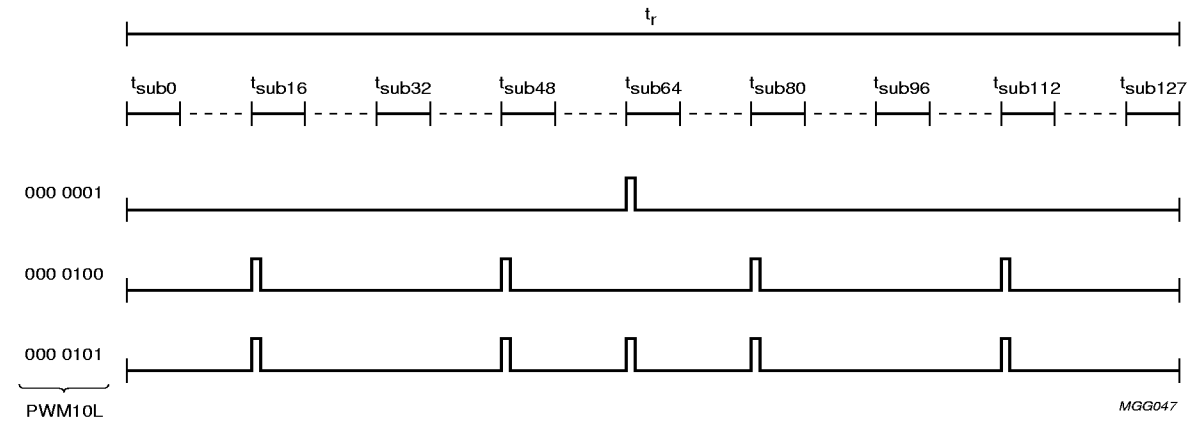


Fig.18 Fine adjustment output (OUT2).

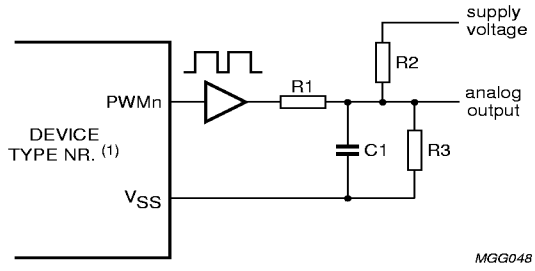
14.3 A typical PWM output application

A typical PWM application is shown in Fig.19. The buffer is optionally used to reduce the influence from supply/ground bouncing to another circuit. R1 and C1 form the integration network, the time constant of which should be equal to or greater than 5 times the repetition period of the PWM output pattern. In order to smooth a changing PWM output a high value of C1 should be chosen. The value of C1 will normally be in the range 1 to 10 μF. The potential divider chain formed by R2 and R3 is used only when the output voltage is to be offset. The output voltages for this application are calculated using Equations (1) and (2).

$$V_{max} = \frac{R3 \times V_{DD}}{R3 + \frac{R1 \times R2}{R1 + R2}} \quad (1)$$

$$V_{min} = \frac{\frac{R1 \times R3}{R1 + R3} \times V_{DD}}{R2 + \frac{R1 \times R3}{R1 + R3}} \quad (2)$$

The loop from the PWM pin through R1 and C1 to V_{SS} will radiate high frequency energy pulses. In order to limit the effect of this unwanted radiation source, the loop should be kept short and a high value of R1 selected. The value of R1 will normally be in the range 3.3 to 100 kΩ. It is good practice to avoid sharing V_{SS} (pin 12) with the return leads of other sensitive signals.



(1) P83Cx80; P87C380

Fig.19 Typical PWM output circuit.

15 ANALOG-TO-DIGITAL CONVERTER (ADC)

The ADC inputs ADC0 and ADC1 share the same pins as port lines P3.2 and P3.3 respectively. Selection of the pin function as either an ADC input or as a port line is achieved using bit ADCE in SFR DFCON (address C0H). When ADCE = 1, the ADC function is enabled; see Section 7.3.2, Table 8.

The two channel ADC comprises a 4-bit Digital-to-Analog Converter (DAC); a comparator; an analog channel selector and control circuitry. As the digital input to the 4-bit DAC is loaded by software (a subroutine in the program), it is known as a software ADC. The block diagram is shown in Fig.20.

The 4-bit DAC analog output voltage (V_{ref}) is determined by the decimal value of the data held in bits DAC0 to DAC3 (DAC value) of SFR ADCDAT (address C1H). V_{ref} is calculated as: $V_{ref} = \frac{V_{DD}}{16} \times (\text{DAC value} + 1)$

Table 31 lists the V_{ref} values as function of DAC3 to DAC0.

When the analog input voltage is higher than V_{ref}, the COMP bit in SFR ADCDAT (address C1H) will be HIGH. The channel selector, consisting of two analog switches, is controlled by bit DACHL in SFR ADCDAT; see Table 32.

Table 31 Selection of V_{ref}

DAC3	DAC2	DAC1	DAC0	V_{ref} (V)
0	0	0	0	$^1/_{16} \times V_{DD}$
0	0	0	1	$^2/_{16} \times V_{DD}$
0	0	1	0	$^3/_{16} \times V_{DD}$
0	0	1	1	$^4/_{16} \times V_{DD}$
0	1	0	0	$^5/_{16} \times V_{DD}$
0	1	0	1	$^6/_{16} \times V_{DD}$
0	1	1	0	$^7/_{16} \times V_{DD}$
0	1	1	1	$^8/_{16} \times V_{DD}$
1	0	0	0	$^9/_{16} \times V_{DD}$
1	0	0	1	$^{10}/_{16} \times V_{DD}$
1	0	1	0	$^{11}/_{16} \times V_{DD}$
1	0	1	1	$^{12}/_{16} \times V_{DD}$
1	1	0	0	$^{13}/_{16} \times V_{DD}$
1	1	0	1	$^{14}/_{16} \times V_{DD}$
1	1	1	0	$^{15}/_{16} \times V_{DD}$
1	1	1	1	V_{DD}

Table 32 Selection of ADC channel

DACHL	CHANNEL SELECTED
0	ADC0
1	ADC1

15.1 Conversion algorithm

There are many algorithms available to achieve the ADC conversion. The algorithm described below and shown in Fig.21 uses an iteration process.

1. Select ADCn channel for conversion. Channel selection is achieved using bit DACHL, SFR ADCDAT (address C1H).
2. Set the digital input to the DAC to 1000. The digital input to the DAC is selected using bits DAC3 to DAC0 (SFR ADCDAT).
3. Determine the result of the compare operation. This is achieved by reading the COMP bit in SFR ADCDAT using the instruction 'MOV A, ADCDAT'. If COMP = 1; the analog input voltage is higher than the reference voltage (V_{ref}). If COMP = 0; the analog input voltage is lower than the reference voltage (V_{ref}).
4. If COMP = 1; then the analog input voltage is higher than the reference voltage (V_{ref}) and therefore the digital input to the DAC needs to be increased. Set the input to the DAC to 1100.
5. If COMP = 0; then the analog input voltage is lower than the reference voltage (V_{ref}) and therefore the digital input to the DAC needs to be decreased. Set the input to the DAC to 0100.
6. Determine the result of the compare operation by reading the COMP bit in SFR ADCDAT.
7. For the DAC = 1100 case.

   If COMP = 1; then the analog input voltage is still greater than V_{ref} and therefore the digital input to the DAC needs to be increased again. Set the input to the DAC to 1110.

   If COMP = 0; then the analog input voltage is now less than V_{ref} and therefore the digital input to the DAC needs to be decreased. Set the input to the DAC to 1010.

8. For the DAC = 0100 case.

   If COMP = 1; then the analog input voltage is now greater than V_{ref} and therefore the digital input to the DAC needs to be increased. Set the input to the DAC to 0110.

   If COMP = 0; then the analog input voltage is still lower than V_{ref} and therefore the digital input to the DAC needs to be decreased again. Set the input to the DAC to 0010.

9. The operations detailed in 5, 6 and 7 above are repeated and each time the digital input to the DAC is changed accordingly; as dictated by the state of the COMP bit.

The complete process is shown in Fig.21. Each time the DAC input is changed the number of values which the analog input can take is reduced by half. In this manner the actual analog value is honed into. The value of the analog input (V_A) is determined by the formulae:

$$V_A = \frac{V_{DD}}{16} \times (\text{DAC value} + 1)$$

As the conversion time of each compare operation is greater than 6 μs but less than 9 μs; a NOP instruction is recommended to be used in between the instructions that select V_{ref}; the ADC channel and read the COMP bit.

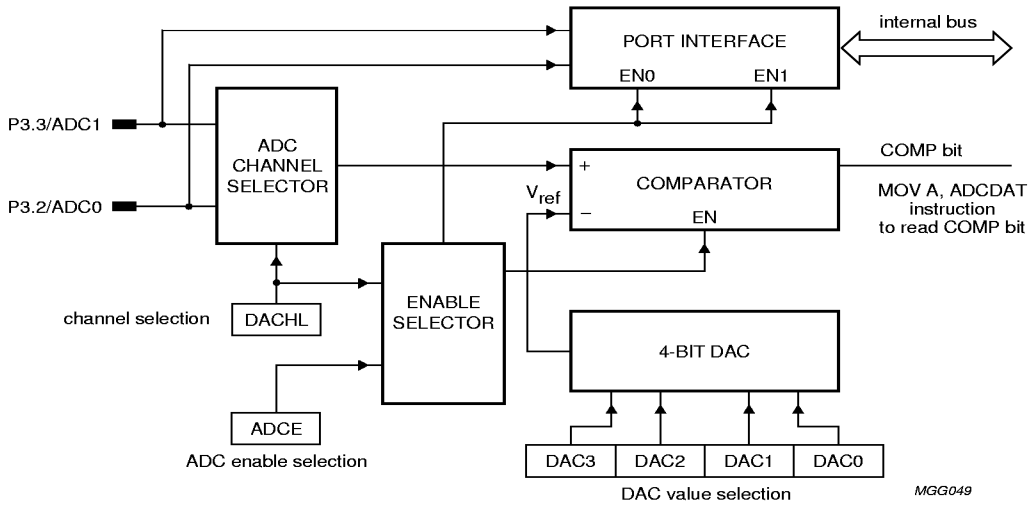


Fig.20 Block diagram of the ADC.

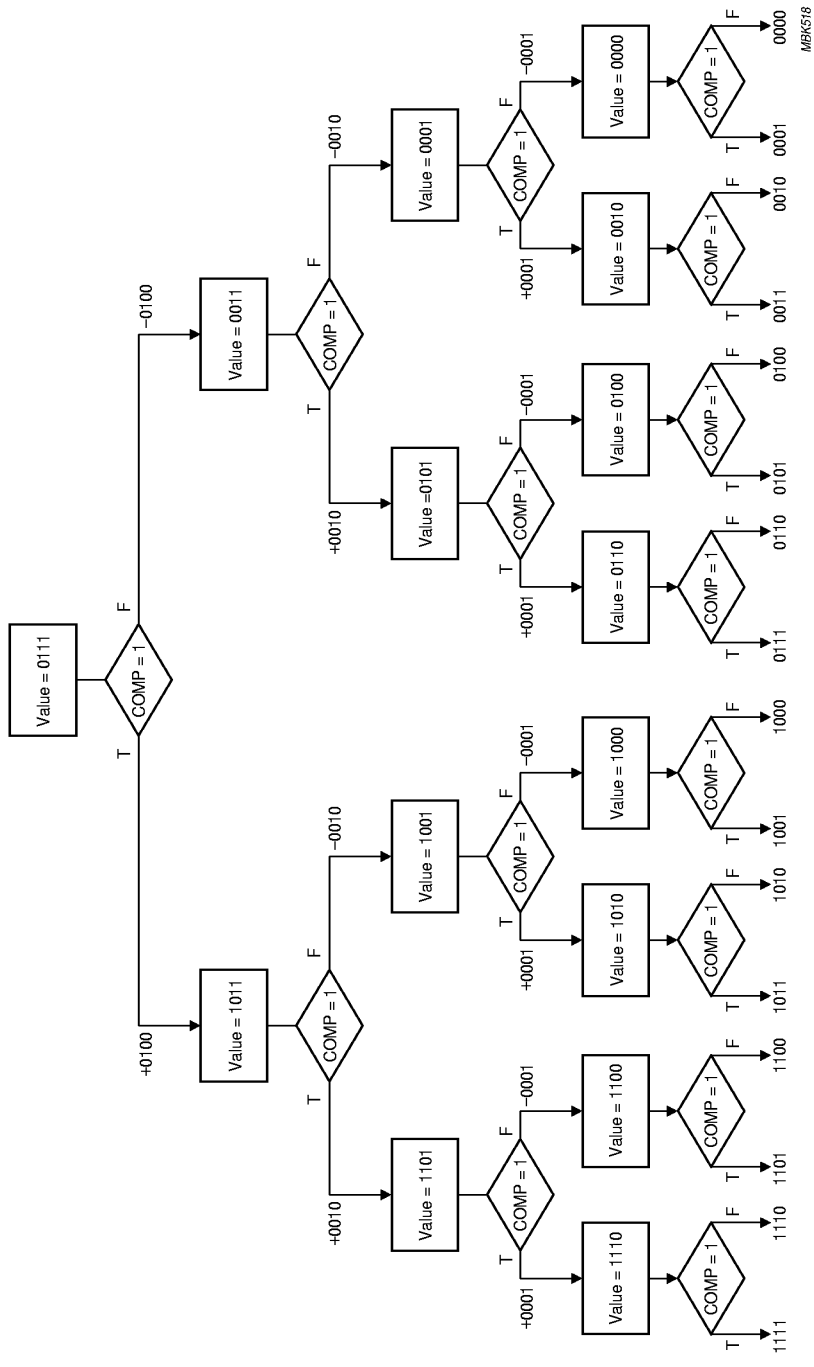


Fig.21 Example of converting algorithm for software ADC.

16 DIGITAL-TO-ANALOG CONVERTER (DAC)

Four channels of linear voltage outputs mainly for the horizontal and vertical position control are provided from four sets of the Digital-to-Analog Converter (DAC).

The DACs are well tuned to meet the following specific system requirements of the monitor.

- For a 21" monitor, the shift per step by changing the data of the DAC is expected to be 0.5 mm. Accordingly, the DAC with 8-bit resolution is required.
- To avoid visible position shift due to environmental temperature change, the drift of output level of the DAC is less than 1.5 mV/°C.
- To eliminate the visible jitter, the maximum tolerable ripple for the output voltage of the DAC is less than 2 mV.

Each DAC comprises an 8-bit data latch, a voltage scaling mechanism and an output driver.

A precise current reference is provided internally to serve 4 sets of DACs. The voltage sources with values weighted by 2^0 up to 2^7 are switched according to the data input so that the sum of the selected voltages gives the required analog voltage from the output driver.

The range of the output voltage is approximately 0 to 5 V. However, to fulfil some special requirement for current driving capability, a very minor reduction (0.2 to 0.4 V) of the output voltage range is acceptable.

The DAC outputs are protected against short-circuits to V_{DDA} and V_{SSA}. To avoid the possibility of oscillation, capacitive loading at the DAC outputs should not exceed 10 pF. In reduced power modes like Idle mode and Power-down mode, the operation of the DACs can be disabled to save the power consumption. In that case, the outputs of the DACs are indeterminate.

Figure 20 illustrates the block diagram of the DAC.

16.1 8-bit Data Registers for the DAC outputs (DACn; n = 0 to 3)

The DACs are individually programmed using an 8-bit word to select an output from one of 256 voltage steps. For V_{DDA} = 5 V, the maximum output voltage of all DACs is 5 V and the resolution is approximately 17.6 mV (5 V/256). At power-on all DAC outputs are set to their lowest value: 00H (i.e. 0 V). The relevant SFRs for the DACs are described in Table 33 and 34.

Table 33 8-bit Data Registers for the DAC outputs (address E9H to ECH)

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

Table 34 Description of DACn bits

BIT	SYMBOL	DESCRIPTION
7 to 0	DACn.7 to DACn.0	8-bit data for the DAC; channel n (n = 0 to 3).

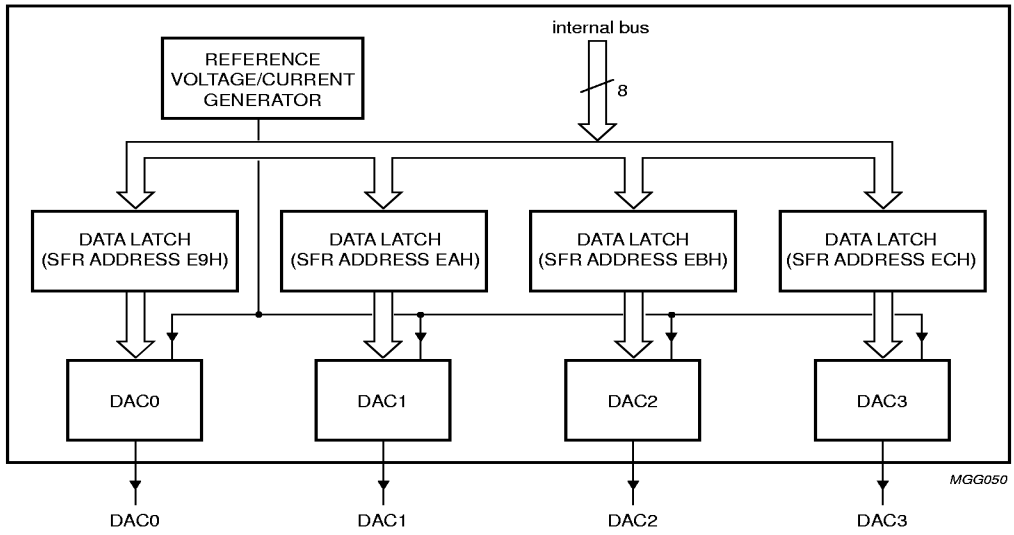


Fig.22 Block diagram of the DAC.

17 DISPLAY DATA CHANNEL (DDC) INTERFACE

The monitor typically includes a number of user controls to set picture size, position, colour balance, brightness and contrast. Furthermore, to optimize some internal setting for different display modes, the timing characteristics should be acquired by the control side. For factory alignment, it is preferable to store the entire information in a non-volatile memory to facilitate production automation. Normally, monitors provide hardware control panels or a keypad to perform this control. However, it is now popular for these controls to go to the PC host. Therefore the communication between monitor and host becomes an issue. DDC1, DDC2B, DDC2B+ and DDC2AB (ACCESS.bus) emerge as a standard for monitor interface.

P83C880 is compliant to DDC1, DDC2B, DDC2B+ and DDC2AB (ACCESS.bus). They also conform to the detection sequence defined by VESA and ACCESS.bus Industry Group and identify the protocol which they are communicating with.

A transmitter clocked by the incoming VSYNC is dedicated for DDC1 operation. An I^2C-bus interface hardware logic forms the kernel of DDC2B and DDC2AB. An address pointer, DDCADR (address 9EH), with post increment capability is employed to serve DDC1, DDC2B, DDC2B+ and DDC2AB modes.

The conceptual block diagram is illustrated in Fig.23.

17.1 Special Function Registers related to the DDC interface

Table 35 SFRs related to the DDC interface

ADDRESS	SFR	REMARKS
9CH	RAMBUF	See Section 7.3.1.
9DH	DDCCON	DDCCON, DDCADR and DDCDAT are mainly created for the DDC1 protocol; they are explained in Sections 17.1.1 to 17.1.3.
9EH	DDCADR	
9FH	DDCDAT	
D8H	S1CON	One extra I^2C-bus interface is used to serve DDC2B, DDC2B+ and DDC2AB. Therefore S1CON, S1STA, S1DAT and S1ADR are just the copies of the corresponding registers in the general I^2C-bus interface. Their usage is exactly the same.
D9H	S1STA	
DAH	S1DAT	
DBH	S1ADR	

17.1.1 DDC Mode Status and DDC1 Control Register (DDCCON)

Table 36 DDC Mode Status and DDC1 Control Register (address 9DH)

7	6	5	4	3	2	1	0
–	EX_DAT	SWENB	–	DDC1_int	DDC1enable	SWH_int	M0

Table 37 Description of DDCCON bits

BIT	SYMBOL	DESCRIPTION
7	–	Reserved.
6	EX_DAT	This bit defines the size of the EDID data. It is related to the function of the post increment of the address pointer, DDCADR. When the upper limit is reached, the address pointer will wrap around to 00H. If EX_DAT = 1, the data size is 256 bytes. If EX_DAT = 0, the data size is 128 bytes; the addressing range for the EDID data buffer is mapped from 0 to 127, the rest (128 to 255) can still be used by the system.
5	SWENB	This bit indicates if the software/CPU is needed to take care of the operation of DDC1 protocol. If SWENB = 1, in DDC1 protocol, the CPU is interrupted during the period of the 9th transmitting bit so that the software service routine can update the hold register of the transmitter by moving new data from the appropriate area (it is not necessary to be the RAM buffer). This transferring must be done within 40 μs. If SWENB = 0, the hold register of the transmitter will be automatically updated from the RAM buffer without the intervention of the CPU.
4	–	Reserved.
3	DDC1_int(1)	Interrupt request bit (002BH is assigned as the interrupt vector address). This bit is only valid in DDC1 protocol while software handling is enabled (SWENB = 1). This bit is set by hardware and should be cleared by software in an interrupt service routine. If DDC1 is fully under the hardware control (SWENB = 0), this bit can be ignored. If DDC1_int = 1, interrupt request is pending. If DDC1_int = 0, there are no interrupt request.
2	DDC1enable(1)	DDC1 enable control bit. If DDC1enable = 1, DDC1 is enabled. If DDC1enable = 0, DDC1 is disabled (the activity on VCLK is ignored).
1	SWH_int(1)	Interrupt request bit (002BH is assigned as the interrupt vector address). This bit is used to indicate that DDC interface switches from DDC1 to DDC2 (i.e. the HIGH-to-LOW transition is observed on pin SCL1). This bit should be cleared by software in an interrupt service routine. If SWH_int = 1, interrupt request is pending. If SWH_int = 0, there is no interrupt request.
0	M0(1)	DDC mode indication bit. If M0 = 0, DDC1 is set; if M0 = 1, DDC2 is set.

Note

1. These bits are R/W.

17.1.2 Address Pointer for DDC Interface Register (DDCADR)

The bits of this register are all R/W.

Table 38 Address Pointer for DDC Interface Register (address 9EH)

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

Table 39 Description of DDCADR bits

BIT	SYMBOL	DESCRIPTION
7 to 0	DDCADR.7 to DDCADR.0	Address pointer with the capability of post increment. After each access to RAMBUF register (either by software or by hardware DDC1 interface), the content of this register will be increased by one. It is available both in DDC1, DDC2 (DDC2B, DDC2B+ and DDC2AB) and system operation.

17.1.3 DDC1 Data Transmission Register (DDCDAT)

Table 40 DDC1 Data Transmission Register (address 9FH)

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

Table 41 Description of DDCDAT bits

BIT	SYMBOL	DESCRIPTION
7 to 0	DDCDAT.7 to DDCDAT.0	Data byte to be transmitted in DDC1 protocol.

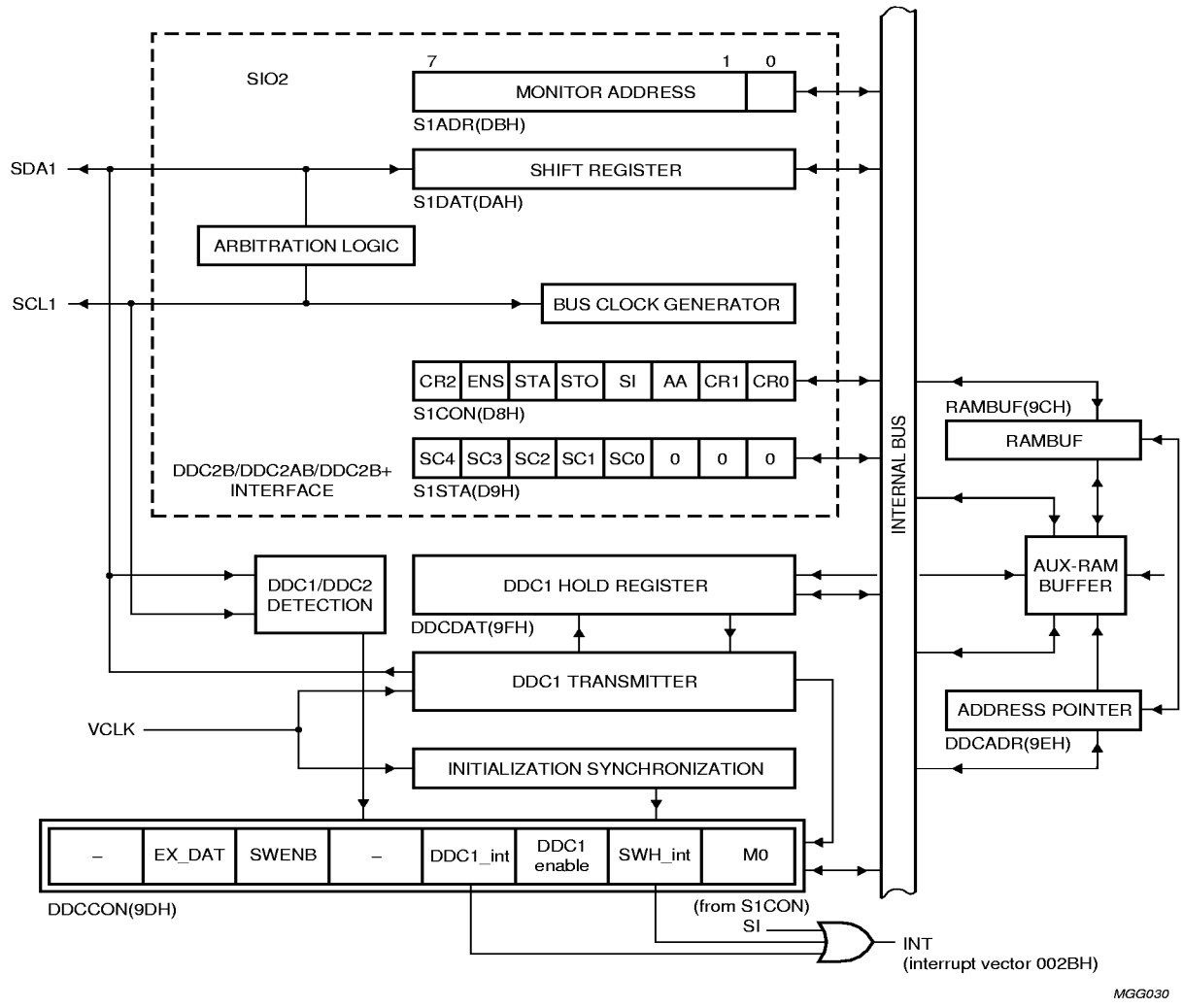


Fig.23 DDC interface block diagram.

17.2 Host type detection

The detection procedure conforms to the sequences proposed by *"VESA Monitor Display Data Channel (DDC) specification"*. The monitor needs to determine the type of host system:

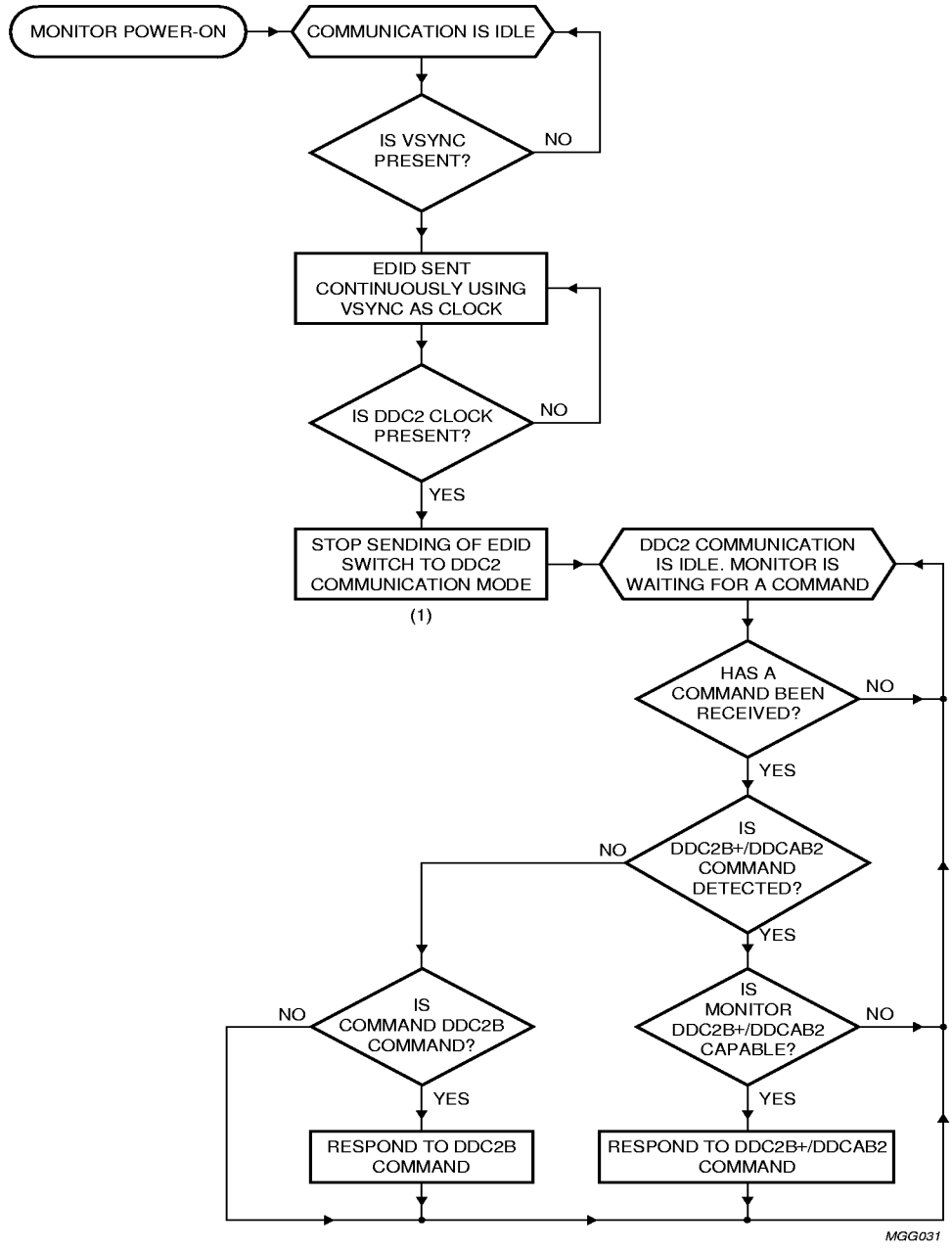
- DDC1 or OLD type host
- DDC2B host (host is master, monitor is always slave)
- DDC2B+/DDC2AB (ACCESS.bus) host.

The sequence of detection is described in the flow chart illustrated in Fig.24.

The monitor where P83C880 resides is always both DDC1 and DDC2 compatible with DDC2 having the higher priority. The display (i.e. P83C880) shall start transmitting DDC1 signals whenever it is switched on and VSYNC is applied to it from the host for the first time. The display shall switch to DDC2 within 3 system clocks as soon as it sees a HIGH-to-LOW transition on the clock line (SCL), indicating that there are both DDC2 devices connected to the bus. Under that condition, the Mode flag M0 will be changed from the default setting logic 0 to logic 1.

Accordingly, the interrupt will be invoked by setting flag SWH_int (DDCCON.1) as HIGH (this flag must be cleared by the interrupt service routine). This procedure will cause a transmission error. However, both the display and the host shall have error detection and a method to recover from the temporary transmission errors.

Figure 24 illustrates the concept and interaction between the monitor and the host. After power-on, the DDC1enable bit (DDCCON.2) is set by software, setting the monitor as a DDC1 device. Therefore, the Mode flag M0, is set as logic 0. Following VSYNC as clock, the monitor (i.e. P83C880) will transmit EDID data stream to the host. However, if DDC2 clock (SCL clock) is present, the monitor will be switched to DDC2B device with the Mode flags setting as logic 1. Software will determine whether it is a DDC2B, DDC2B+, or DDC2AB protocol.



(1) Once the monitor switches from DDC1 mode to DDC2 mode it will remain in DDC2 mode for the duration of that power-on period.

Fig.24 Host type detection.

17.3 DDC1 protocol

The DDC1 is a primitive interface, but adopted by many monitor models and PC hosts. It is a point-to-point interface. The monitor is always set to 'Transmit only' mode. In the initialization phase, 9 clock cycles on the VCLK pin will be given for the internal synchronization. During this period, the SDA pin will be kept in the high-impedance state. By default, bit 'DDC1enable' is reset as logic 0. It is advised to move the EDID data to the RAM buffer before enabling DDC1. To activate the DDC1 interface, DDC1enable flag is set to logic 1 and it is taken as granted that Mode flag M0 is set at logic 0. If SWENB is kept at the default value logic 0, the RAM buffer will be tied to the DDC1 protocol. The allocated size from the RAM buffer is decided by the flag EX_DAT. If EX_DAT is LOW, only 128 bytes are reserved to store DDC1 EDID data. The upper part (locations 128 to 255) is still available to the system. If EX_DAT is HIGH, the entire 256 bytes of the RAM buffer are dedicated to the usage of DDC1 operation.

The hardware mechanism will automatically move new data from the defined RAM buffer to the hold register of the transmitter with the aid of the address pointer DDCADR. Within the range of DDC1 RAM buffer, the function of the post increment is executed. If the upper limit is reached, the address pointer DDCADR will wrap around to 00H. However, if EX_DAT = LOW, the lower part is occupied by the DDC1 operation and the upper part is still free to the system. Nevertheless, the effect of the post increment just applies to the part related to the DDC1 operation. In other words, the system program is still able to address the locations from 128 to 255 in the RAM buffer through the MOVX command but without the facility of the post increment; e.g. in case EX_DAT = LOW and SWENB = LOW, the system program might read one data byte from address 200 of the RAM buffer by the following procedure:

MOV R0, #200; MOVX A, @R0.

The address pointer DDCADR is tied to the DDC1 transmitter here. To avoid the interference to the content of DDCADR, it has to address the RAM buffer through the MOVX command. While EX_DAT = HIGH, the entire RAM buffer is covered by the pointing range of the address pointer DDCADR, with the capability of the post increment; no matter whether the access is done by DDC1 related hardware (SWENB = 0) or software (SWENB = 1).

If SWENB is set at logic 1, then after the valid synchronization, the DDC1 interrupt will be invoked (interrupt vector address 002BH). The service routine should fill the hold register DDCDAT (SFR address 9FH) with the first data byte either from the internal ROM (part of the system ROM) or from the RAM buffer. In the latter case, the address pointer DDCADR will provide the benefit of the post increment for the service routine to read/write the DDC1 EDID data area. This action must be finished within 40 μs (40 machine cycles in 12 MHz system clock). DDC1_int flag must be cleared by software before it returns from service routine. On the rising edge of the 10th clock cycle, the device will output the first valid data bit which should be the most significant bit of a byte.

The following data is also transmitted on the SDA pin in 8 bits per byte format. Each byte is followed by a 9th clock pulse during which time SDA is left high-impedance and either the hardware mechanism (SWENB = 0) or the service routine (SWENB = 1) will update the hold register. The data bit is output on the rising edge of VCLK, the most significant bit first. The address pointer is initialized at 00H. After writing a data byte to the hold register through hardware or software, it will be incremented by one automatically. If the address reaches 127 (EX_DAT = 0) or 255 (EX_DAT = 1) the address pointer will wrap around to the first location: 00H. Nevertheless, it is possible for CPU (in software mode, i.e. SWENB = 1) to write any desired address to the address pointer to proceed random access. The transaction in DDC1 protocol is shown in Fig.25.

If DDC1 hardware mode is used, the following DDC1 operation steps are recommended:

1. Reset DDC1enable (by default DDC1enable is cleared to LOW after power-on reset).
2. Set SWENB to HIGH.
3. Depending on the data size of EDID data, set EX_DAT to LOW (128 bytes) or HIGH (256 bytes).
4. Use substantial moving commands (DDCADR, RAMBUF involved) to move the entire EDID data to RAM buffer.
5. Reset SWENB to LOW.
6. Reset DDCADR to 00H.
7. Set DDC1enable to HIGH enabling the DDC1 hardware; during the synchronization phase (the first 9 VCLK clocks) the first data byte will be loaded into the shift register of the transmitter through the hold register.

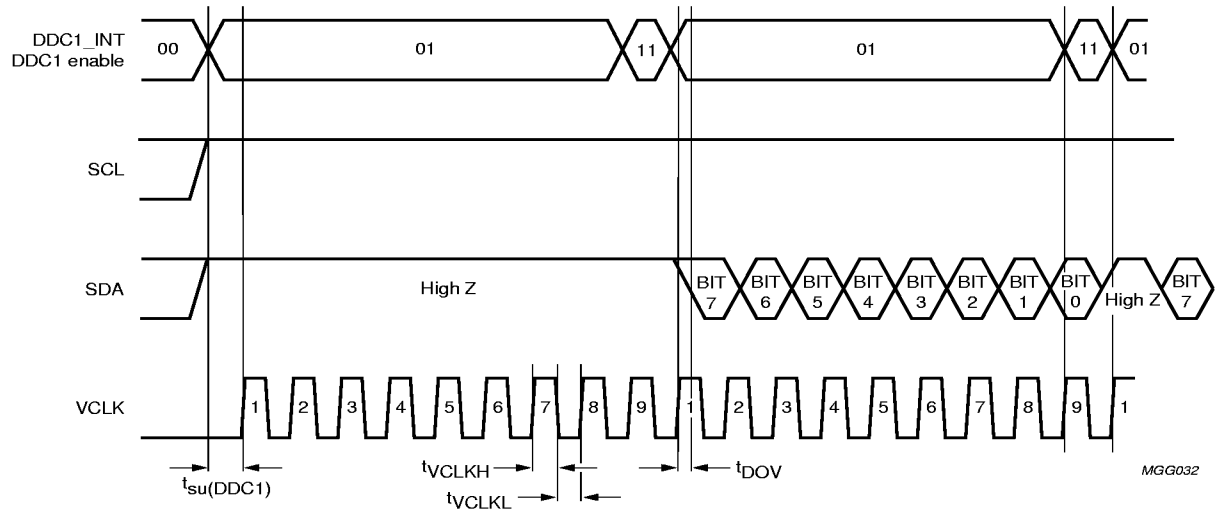


Fig.25 Transmission protocol in DDC1 interface.

17.4 DDC2B protocol

The DDC2B construction is based on the Philips I²C-bus interface. However, in the level of DDC2B, PC host is fixed as the master and the monitor is always regarded as the slave. Both master and slave can be operated as a transmitter or receiver, but the master device determines which mode is activated. For details of the I²C-bus interface, please refer to the Philips publication *"The I²C-bus and how to use it"* ordering number 9398 393 40011 and/or the *"Data Handbook IC20"*.

In the P83C880, one more pair of I²C-bus pins SCL1, SDA1 and an I²C-bus hardware interface logic are dedicated to perform DDC2B/DDC2B+/DDC2AB protocols. The built-in address pointer mentioned in Section 17.3 can be used to speed up the processing of service routines for the access of the internal ROM or a dedicated RAM buffer.

According to the DDC2B specification:

- A0H (for write mode) and
- A1H (for read mode),

are assigned as the default address of monitors.

The reception of the incoming data in write mode or the updating of the outgoing data in read mode should be finished within the specified time limit. However, it is not necessary for EDID data to be stored on-chip. It is also possible to have access to the external EEPROM/ROM through another set of I²C-bus interface and pre-store those data in the RAM buffer. If software on the slave side cannot react to the master in time, based on I²C-bus protocol, SCL pin can be stretched LOW to inhibit the further action from the master. The transaction can be proceeded in either byte or burst format.

17.5 DDC2AB/DDC2B+ protocol

DDC2AB/DDC2B+ is a superset of DDC2B. Monitors that implement DDC2AB/DDC2B+ are full featured ACCESS.bus devices. Essentially, they are similar to DDC2B. The I²C-bus interface forms the fundamental layer for both protocols. However, in DDC2AB/DDC2B+ the default address for monitors is assigned to 6EH instead of A0H and A1H in DDC2B. Monitors and hosts can play both the roles of master and slave. Under this kind of protocol, it is easy to extend the support for hosts to read VESA Video Display Information Format (VDIF) and remotely control monitor functions.

The read/write protocols can be the same as described in Section 17.4.

Nevertheless, the command/information sequence between host and monitor must conform to the specification of ACCESS.bus.

Timing rules specified in ACCESS.bus such as maximum response time to RESET message (<250 ms) from host, maximum time to hold SCL LOW (<2 ms) etc., can be satisfied through software checks and built-in timers such as Timer 0 and Timer 1 in the P83C880. In DDC2AB/DDC2B+ the monitor itself can act as a master to activate the transaction. The default address assigned for the host is 50H.

Figure 26 shows the overview and relationship between software and the existing I²C-bus hardware for the DDC interface.

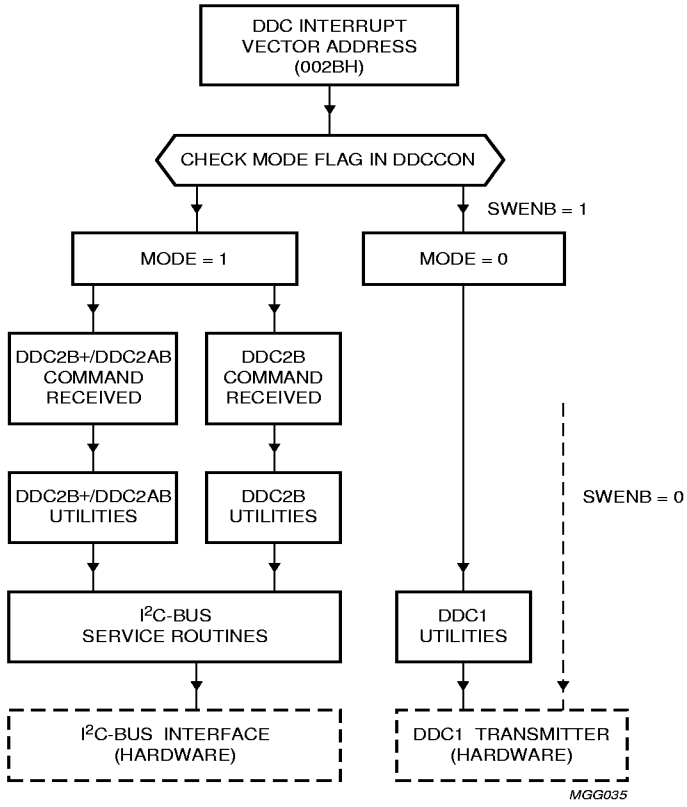


Fig.26 The conceptual structure of the DDC interface.

17.6 RAM buffer for the system and DDC application

The architecture of the RAM buffer is set up in a flexible configuration to use the RAM resource to a maximum. In principle the RAM buffer can be shared as system or DDC RAM buffer. The relationship among those applications is arranged as shown in Table 42.

Table 42 Relation between system RAM and DDC RAM

MODE	EX_DAT	SWENB	AUX RAM 0 TO 127		AUX RAM 128 TO 255		
			NORMALLY RESERVED FOR	NOTE	NORMALLY RESERVED FOR	AVAILABLE FOR	NOTE
DDC1	0	0	DDC1 EDID data	1, 2 and 3	–	system access	2
	0	1		3 and 4			–
	1	0		1 and 2	DDC1 EDID data	–	1, 2 and 5
	1	1		4			4 and 5
DDC2	0	1	DDC2 EDID data	3	–	system access	–
	1	1		–	DDC2 EDID data	–	5

Notes

1. Read/write through MOVX instruction might conflict with the access from DDC1 hardware. So the access from CPU by using MOVX instruction is forbidden.
2. READ/WRITE through DDCADR and RAMBUF registers has the conflicting problem also. Even the content of DDCADR, which should be employed by DDC1 hardware, will be damaged. So it is inhibited to use this type of access.
3. If DDCADR reaches 127 it will automatically wrap around to 0 after the access is done.
4. The access conflict can be avoided because DDC1 access is done by the interrupt service routine. However, the EDID transferring from the RAM buffer should be finished within 40 μs.
5. If DDCADR reaches 255 it will automatically wrap around to 0 after the access is done.

18 I²C-BUS INTERFACE

The P83C880 has a software I²C-bus interface that can be used in master mode. Full details of the I²C-bus are given in the 80C51 family *"Data Handbook IC20"* and/or the document *"The I²C-bus and how to use it"* (ordering number 9398 393 40011).

The I²C-bus interface lines SDA and SCL share the same pins as Port 1 lines P1.1 and P1.0 respectively; selection is done via the S1E bit in SFR DFCON (address C0H).

19 HARDWARE MODE DETECTION

Due to the many restrictions for mode detection via software, a hardware mode detector is introduced in the P83C880. This feature has the following advantages:

- Fast enough to react on any mode change.
- Possibility to detect the polarity of HSYNC without extra input pin.
- Higher accuracy to measure the HSYNC and VSYNC frequency
- Relieves the CPU and valuable timers/counters in the microcontroller from lengthy mode detection
- In reduced power modes, e.g. Idle mode, the mode detection is still active.

Apart from the above mentioned benefits, some extra features can be achieved through hardware mode detection. The composite sync separation can be done while mode detection continues. The various formats of HSYNC and VSYNC output pulses are provided to fit into different applications.

Since the information of the operational modes defined in Device Power Management Signalling (DPMS) for monitors is embedded in HSYNC and VSYNC, the mode detector is also able to recognize the operational mode and generate the fixed free running frequency for HSYNC and VSYNC in some specific modes like stand-by, suspend and power-off. Two display patterns, the white and the cross hatch, can be displayed for the self test in the free running mode.

More on Device Power Management Signalling (DPMS) will be explained in Chapter 20, “Power management”.

19.1 Special Function Register for mode detection and sync separation

There are 4 SFRs related to mode detection and sync separation: MDCST (SFR address F8H); HFHIGH (SFR address FCH); VFHIGH (SFR address FDH) and VFLHFL (SFR address FEH).

Five SFRs are applied for the output pulse generation of HSYNC and VSYNC and the display pattern generation for the self test: HFP (SFR address F6H); HFPOPW (SFR address F7H); VFP (SFR address F9H); VFPOPW (SFR address FAH) and PULCNT (SFR address FBH).

These SFRs are described in detail in Section 19.1.1 to 19.1.9.

19.1.1 Mode Detection/sync separation Control and Status Register (MDCST)

Table 43 Mode Detection/sync separation Control and Status Register (SFR address F8H)

7	6	5	4	3	2	1	0
MARCH	CHREQ	XSEL	HSEL	Hpres	Vpres	Hpol	Vpol

Table 44 Description of MDCST bits

BIT	SYMBOL	DESCRIPTION
7	MARCH	**Mode detection enable**. If MARCH = 1, mode detection is enabled. If MARCH = 0, mode detection is disabled. Default value MARCH = 1 to guarantee that mode detection is automatically executed after power-on reset.
6	CHREQ	Polarity or frequency change. If CHREQ = 1 then the polarity or frequency change happened. The polarity or frequency is detected after a CHREQ HIGH-to-LOW transition, because the mode change becomes stable after the CHREQ HIGH-to-LOW transition.
5	XSEL	Indicates if the clock used for the mode detection and the pulse generation is $\frac{1}{2} \times f_{clk}$. If XSEL = 1, the internal clock is $\frac{1}{2} \times f_{clk}$. If XSEL = 0 the internal clock is equal to f_{clk}.
4	HSEL	Indicates the horizontal sync input coming from $HSYNC_{in}$ (pin $HSYNC_{in}/\overline{PROG}$) or $CSYNC_{in}$ (pin $CSYNC_{in}$/P1.6). If HSEL = 1, $CSYNC_{in}$ is the input pin. If HSEL = 0, $HSYNC_{in}$ is the input pin.
3	Hpres	Indicates the presence of HSYNC. If Hpres = 1, HSYNC is present. If Hpres = 0, HSYNC is not present.

BIT	SYMBOL	DESCRIPTION
2	Vpres	Indicates the presence of VSYNC. If Vpres = 1, VSYNC is present. If Vpres = 0, VSYNC is not present.
1	Hpol	Indicates the polarity of HSYNC. If Hpol = 0, polarity is positive (active HIGH; i.e. HIGH at horizontal retracing period). If Hpol = 1, polarity is negative (active LOW; i.e. LOW at horizontal retracing period).
0	Vpol	Indicates the polarity of VSYNC. If Vpol = 0, polarity is positive (active HIGH; i.e. HIGH at vertical retracing period). If Hpol = 1, polarity is negative (active LOW; i.e. LOW at vertical retracing period).

19.1.2 Horizontal Period Counting High-byte Register (HFHIGH)

Table 45 Horizontal Period Counting High-byte Register (SFR address FCH)

7	6	5	4	3	2	1	0
HF.11	HF.10	HF.9	HF.8	HF.7	HF.6	HF.5	HF.4

Table 46 Description of HFHIGH bits

BIT	SYMBOL	DESCRIPTION
7 to 0	HF.11 to HF.4	Indicate the high byte of the value counted by mode detector for 4 horizontal scanning periods.

19.1.3 Vertical Period Counting High-byte Register (VFHIGH)

Table 47 Vertical Period Counting High-byte Register (SFR address FDH)

7	6	5	4	3	2	1	0
VF.11	VF.10	VF.9	VF.8	VF.7	VF.6	VF.5	VF.4

Table 48 Description of VFHIGH bits

BIT	SYMBOL	DESCRIPTION
7 to 0	VF.11 to VF.4	Indicate the high byte of the value counted by mode detector for 1 vertical field/frame period.

19.1.4 Vertical and Horizontal Period Counting Low-nibbles Register (VFLHFL)

Table 49 Register containing the low nibbles of the Horizontal and Vertical Period Counting (SFR address FEH)

7	6	5	4	3	2	1	0
VF.3	VF.2	VF.1	VF.0	HF.3	HF.2	HF.1	HF.0

Table 50 Description of VFLHFL bits

BIT	SYMBOL	DESCRIPTION
7 to 4	VF.3 to VF.0	Indicate the low nibble of the value counted by mode detector for 1 vertical field/frame period.
3 to 0	HF.3 to HF.0	Indicate the low nibble of the value counted by mode detector for 4 horizontal scanning periods.

19.1.5 Horizontal free running Frequency/Period byte Register (HFP)

Table 51 Horizontal free running Frequency/Period Register byte Register (SFR address F6H)

7	6	5	4	3	2	1	0
HFP9	HFP8	HFP7	HFP6	HFP5	HFP4	HFP3	HFP2

Table 52 Description of HFP bits

BIT	SYMBOL	DESCRIPTION
7 to 0	HFP9 to HFP2	Indicate the upper byte of the 10 bits for programming the free running HSYNC output pulse.

19.1.6 Horizontal free running Frequency/Period and Pulse Width Register (HFPOPW)

Table 53 Horizontal free running Frequency/Period and Pulse Width Register (SFR address F7H)

7	6	5	4	3	2	1	0
VSEL	HFP1	HFP0	HOPW4	HOPW3	HOPW2	HOPW1	HOPW0

Table 54 Description of HFPOPW bits

BIT	SYMBOL	DESCRIPTION
7	VSEL	Indicates whether the internal vertical sync is coming from the external VSYNC input pin ($VSYNC_{in}/\overline{OE}$) or separated from the composite signal. If VSEL = 0, vertical sync coming from the external VSYNC input pin. If VSEL = 1, vertical sync separated from the composite signal.
6 to 5	HFP1 to HFP0	Indicate the lower two of the 10 bits for programming the free running horizontal output pulse.
4 to 0	HOPW4 to HOPW0	Indicate the horizontal output pulse width.

19.1.7 Vertical free running Frequency/Period byte Register (VFP)

Table 55 Vertical free running Frequency/Period byte Register (SFR address F9H)

7	6	5	4	3	2	1	0
VFP9	VFP8	VFP7	VFP6	VFP5	VFP4	VFP3	VFP2

Table 56 Description of VFP bits

BIT	SYMBOL	DESCRIPTION
7 to 0	VFP9 to VFP2	Indicate the upper byte of the 10 bits for programming the free running vertical output pulse.

19.1.8 Vertical free running Frequency/Period and Pulse Width Register (VFPOPW)

Table 57 Vertical free running Frequency/Period and Pulse Width Register (SFR address FAH)

7	6	5	4	3	2	1	0
–	–	VFP1	VFP0	VOPW3	VOPW2	VOPW1	VOPW0

Table 58 Description of VFPOPW bits

BIT	SYMBOL	DESCRIPTION
7 to 6	–	Reserved.
5	VFP1	Indicate the lower two bits of the 10 bits for programming the free running vertical output pulse.
4	VFP0	
3 to 0	VOPW3 to VOPW0	Indicate the vertical output pulse width.

19.1.9 Pulse generation Control Register (PULCNT)

Table 59 Pulse generation Control Register (SFR address FBH)

7	6	5	4	3	2	1	0
CLMPEN	PATTYP	VPG	PVSI	HPG1	HPG0	FBPO	PHSI

Table 60 Description of PULCNT bits

BIT	SYMBOL	DESCRIPTION
7	CLMPEN	**Clamping pulse output (CLAMP) enable**. If CLMPEN = 1, pin PWM8/CLAMP/P3.0 is switched to the CLAMP output. If CLMPEN = 0, the CLAMP function is disabled. The CLAMP function always overrides other alternative functions such as PWM8 and P3.0.
6	PATTYP	**Basic display patterns selection**. If PATTYP = 0, the white display pattern is selected. If PATTYP = 1, the cross hatch display pattern is selected.
5	VPG	**Vertical pulse output modes selection**. If VPG = 0, a free running vertical sync pulse signal is selected. If VPG = 1, a vertical substitution pulse signal is selected.
4	PVSI	**Vertical output pulse polarity selection**. If PVSI = 0, the positive polarity is selected. If PVSI = 1, the negative polarity is selected.
3 to 2	HPG1 to HPG0	**Horizontal pulse output modes selection**; see Table 61.
1	FBPO	**The clamp pulse at the back porch or at the front porch selection**. This bit is only valid when CLMPEN is set HIGH. If FBPO = 1, the horizontal back porch clamp pulse is selected. If FBPO = 0, the horizontal front porch clamp pulse is selected.
0	PHSI	**Polarity of the horizontal and clamping output pulse indication.** If PHSI = 0, the positive polarity is selected. If PHSI = 1, the negative polarity is selected.

Table 61 Horizontal pulse output modes

HPG1	HPG0	MODES
0	0	free running horizontal sync pulse signal
0	1	horizontal substitution pulse signal
1	0	reserved
1	1	the incoming horizontal sync is followed and delivered out but the horizontal substitution pulse is disabled, while the incoming HSYNC is missing

19.2 System description

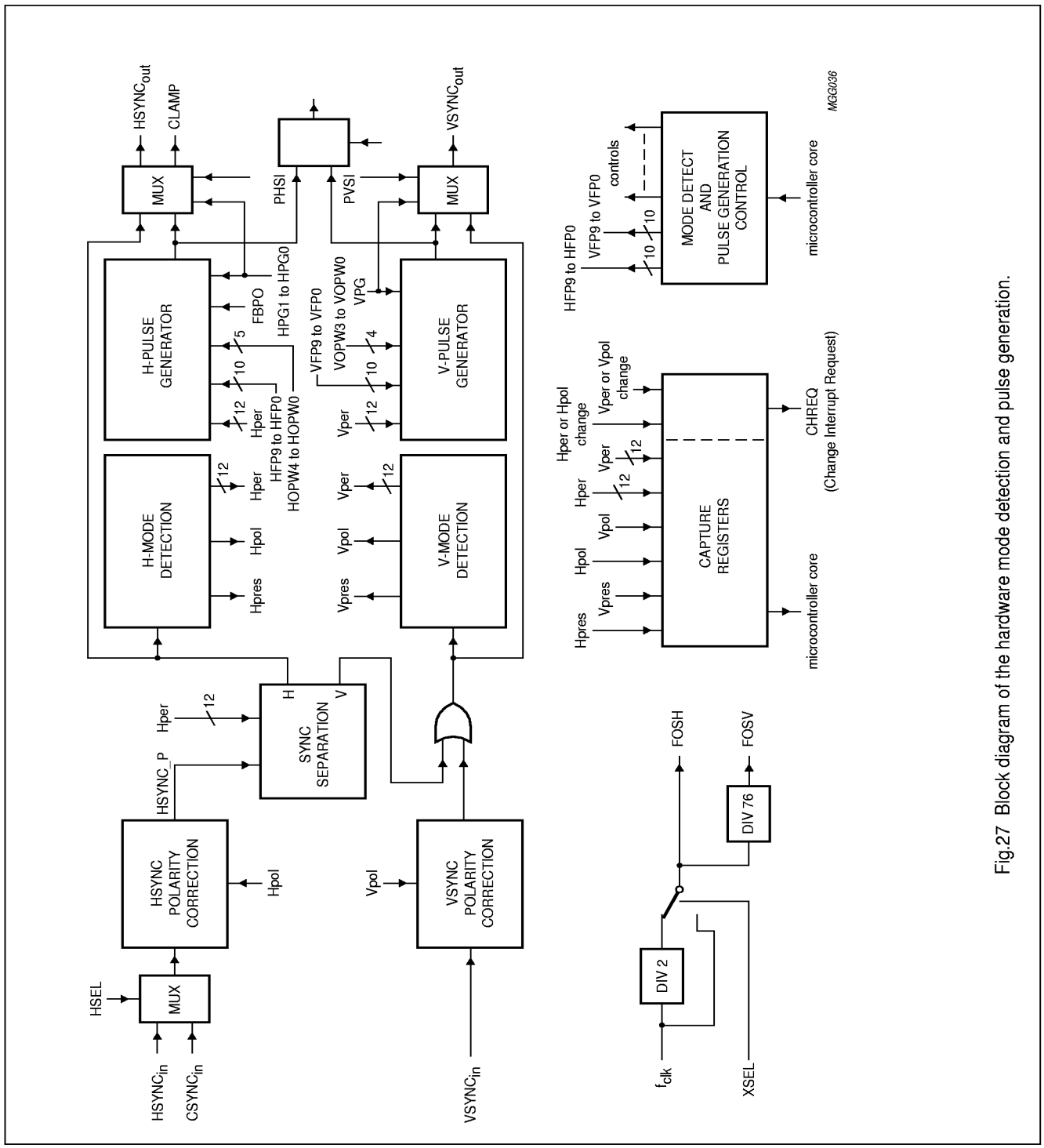


Fig.27 Block diagram of the hardware mode detection and pulse generation.

19.2.1 Clock prescaler

To reach the required 12 bit accuracy and the reasonable crystal clock frequency, the line time of 4 consecutive lines is measured (see Section 19.2.5). Doing so the maximum crystal clock frequency is:

$$f_{clk} \leq \left(\frac{2^{12} \times f_{H(min)}}{4} = 15.36\ MHz \right)$$

From the calculation above it is clear that there will be no overflow if f_{clk} is 10 or 12 MHz. However, for a 16 or 24 MHz crystal f_{clk} has to be divided by a factor of 2 to get a correct clock frequency FOSH for the horizontal part of the mode detection. If f_{clk} used is 10, 12 or 16 MHz then FOSH will be respectively 10, 12 or 8 MHz.

The same holds for the vertical part of the mode detection. Here the minimum vertical sync frequency $f_{V(min)}$ = 40 Hz, so the maximum clock frequency to avoid overflow in a 12-bit counter, equals:

$$FOSV \leq \left(2^{12} \times f_{V(min)} = 163.84\ kHz \right)$$

To get a maximum accuracy without overflow the clock of the horizontal section FOSH, should be prescaled down to this value of FOSV. Thus the scale factor should be at

$$least: n \geq \left(\frac{FOSH_{(max)}}{163.84\ kHz} = \frac{12 \times 10^3\ kHz}{163.84\ kHz} = 73.2 \right)$$

Take n = 76 as a suitable scale factor as shown in Fig.23.

Table 62 Clock frequencies

f_{clk} (MHz)	FOSH (MHz)	FOSV (MHz)
10	10	131.6
12	12	157.9
16	8	105.3

19.2.2 Horizontal polarity correction

In order to simplify the processing in the following stages, the HSYNC polarity correction circuit is able to convert the input sync signals to positive polarity signals in all situations. However, be aware that this correction is achieved by the aid of Hpol and Hper signals. Hpol and Hper are only settled down in several horizontal scanning lines (61 lines, if f_{HSYNC} goes up) or a few milliseconds (worst case 1.2 ms, if f_{HSYNC} becomes inactive) after power-on or timing mode change.

19.2.3 Vertical polarity correction

The purpose of the vertical polarity correction is similar to the horizontal polarity correction. However, it takes a longer time to get the correct result after power-on or a timing mode change because at least 5 frames are needed to stabilize Vper from the input sync signals.

19.2.4 Vertical sync separation

This function will separate the vertical sync out of the composite sync. To do so the change in polarity during VSYNC interval can be utilized to extract VSYNC as shown in Fig.28. The differentiated sync pulse derived from HSYNC_P is used to reset an 8-bit upcounter. At $\frac{1}{4}$ of the line time the level of the incoming sync at that moment is clocked into the last D flip-flop. Be aware that $\frac{1}{4}$ of the line period equals $\frac{1}{16}$ of parameter Hper because Hper is calculated based on 4 consecutive lines.

Due to the fact that the differentiator reacts upon every LOW-to-HIGH transition even an interlaced composite sync will be separated correctly. Note further that the sync separation is taken from signal HSYNC_P which is processed by the Horizontal Polarity Correction circuit but not necessary with the fixed polarity. As a result the polarity of the separated vertical sync Vsep will vary with the incoming composite signal, and in case no composite sync is present the level will be LOW. According to this algorithm, Vsep will be always $\frac{1}{4}$HSYNC period shift to the original VSYNC window. The 8-bit counter in Fig.28 is stopped at its maximum of FFH, otherwise it will start again from zero which results in a wrong enable pulse for the D flip-flop.

All kinds of composite sync input shown in Chapter 27 can be dealt with by this function. There are 6 types of composite sync waveforms which can be accepted by the sync processor (see Fig.39).

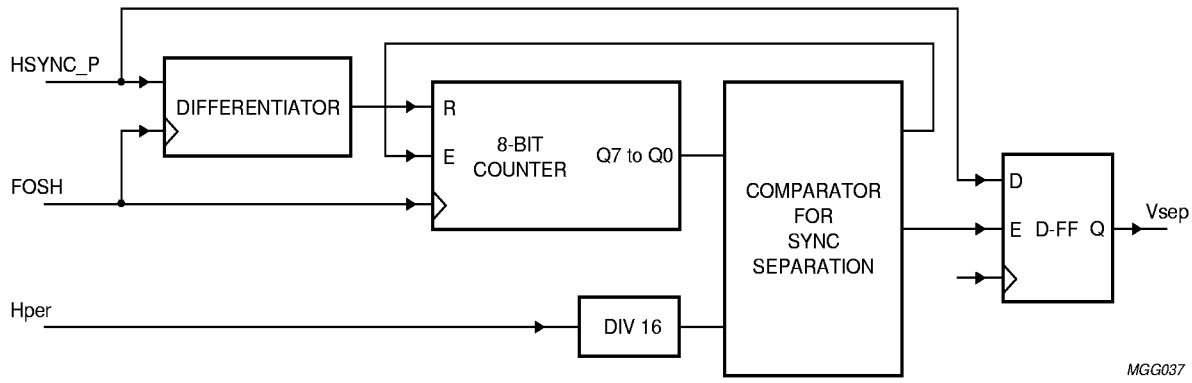


Fig.28 Vertical sync separator.

19.2.5 HORIZONTAL MODE DETECTION

This function block determines the following 4 parameters: Hper, Hpol, Hpres and Hcha. The functional block diagram is shown in Fig.29.

The 12-bit counter for the determination of the horizontal period has a synchronous reset and will be reset by the prescaled and differentiated horizontal pulse HSYNC_P, coming from the horizontal prescaler, or in case the counter reaches the value FF0H as defined by the comparator. The latter situation will occur if no sync pulses are coming in or if the frequency of the incoming sync pulses is too low.

The 12-bit register, that will store the 4 line time period, is only enabled if the signal HENA is HIGH.

19.2.5.1 Parameter Hper

Based on this configuration, the resulting accuracy for Hper will be:

$$\text{Accuracy in H} = \frac{100\%}{(\text{horizontal counter value})} = \frac{100\% \times f_H}{4 \times \text{FOSH}}$$

Table 63 Accuracies

XTAL (f_{clk}) (MHz)	FOSH (MHz)	ACCURACY IN H AT 30KHZ (%)	ACCURACY IN V AT 50HZ (%)
10	10	0.075	0.038
12	12	0.063	0.032
16	8	0.094	0.048

A 12-bit counter is used to measure the frequency of HSYNC. Every 4 HSYNC periods are measured and compared with the previous locked HSYNC frequency; the counter is reset every 4 HSYNC periods. The base clock to the counter is $\frac{1}{12\ \text{MHz}} = 83$ ns for a 12 MHz system clock. If the new HSYNC frequency is larger than the previous HSYNC frequency, there is a 4-bit counter to avoid the noise or the temporary frequency shift. When the 4-bit counter counts up to 15 for the comparison $\text{Hfreq}_{(\text{new})} > \text{Hfreq}_{(\text{previous})}$, then the mode change interrupt is issued. The time from mode change to interrupt issued is approximately $4 \times \text{Hperiod}_{(\text{new})} \times 15$ or $(4 \times \text{Hperiod}_{(\text{new})} \times 14) + n \times \text{Hperiod}_{(\text{previous})}$, where n = 1, 2 or 3.

If the new HSYNC frequency is not in the static state and is smaller than the previous HSYNC frequency, the time from mode change to interrupt issued is approximately $(4 \times \text{Hperiod}_{(\text{new})} \times 3) - (n \times \text{Hperiod}_{(\text{previous})})$, where n = 1, 2, 3 or 4. The static state is considered as the state where the 12-bit counter, for measuring the HSYNC frequency, reaches 4080.

For a 12 MHz system clock, the static state is equal to

$$\frac{1}{83\ \text{ns} \times \frac{4080}{4}} = \frac{1}{84660\ \text{ns}} = 11.8\ \text{kHz}.$$

If the input HSYNC frequency is less than 11.8 kHz, it will be regarded as static state. A 2-bit counter is used to avoid the noise or the temporary shift.

When the 2-bit counter counts up to 3 for the comparison 'static state' < $\text{Hfreq}_{(\text{new})}$ < $\text{Hfreq}_{(\text{previous})}$, then the mode change interrupt is issued.

When the new input HSYNC is changed to a static state, the HSYNC frequency counter reaches 4080. In order to have a faster response of the mode change interrupt, the 2-bit counter is not used and the mode change interrupt is issued immediately. The time from mode change to interrupt issued is approximately $((4080 + 5) \times 83\ \text{ns}) - (n \times \text{Hperiod}_{(\text{previous})})$, where n = 1, 2, 3 or 4.

19.2.5.2 Parameter Hpres

The presence indicator Hpres, as stored in the SR flip-flop, contains some hysteresis, as required due to the two threshold settings in the comparator. The SR flip-flop will be set, indicating that no active sync (or sync with a too low frequency) is coming in if the counter reaches a value of FF0H (4080 decimal). The corresponding horizontal sync frequency is:

$$f_H = \text{Hfreq1} = \frac{4 \times \text{FOSH}}{(\text{Counter value})} = \frac{4 \times \text{FOSH}}{4080}$$

The SR flip-flop will be reset, indicating that an active sync is present, if the counter reaches a value lower than FC0H (4032 decimal). The corresponding horizontal sync frequency is:

$$f_H = \text{Hfreq2} = \frac{4 \times \text{FOSH}}{(\text{Counter value})} = \frac{4 \times \text{FOSH}}{4032}$$

The hysteresis result,

$$\text{Hysteresis} = (\text{Hfreq1} - \text{Hfreq2}) / \text{Hfreq1},$$

is shown in Table 64.

Table 64 Hysteresis in the Hpres detection

XTAL (f_{clk}) (MHz)	FOSH (MHz)	Hfreq1 (MHz)	Hfreq2 (MHz)	HYSTERESIS (%)
10	10	9.804	9.921	1.2
12	12	11.765	11.905	1.2
16	8	7.843	7.937	1.2

19.2.5.3 Parameter Hpol

The mode change interrupt will be issued after 36 to 40 periods of the HSYNC when the polarity is changed from negative to positive. The polarity indicator Hpol will be changed from a logic 1 to a logic 0. The mode change interrupt will be issued after the 60 periods of the HSYNC when the polarity is changed from positive to negative. The Hpol will be changed from a logic 0 to a logic 1. A logic 0 represents a positive polarity and a logic 1 represents a negative polarity.

19.2.5.4 Parameter Hcha

If Hcha is set to HIGH indicating that the period time has changed for a long period, the Hper and Hpres will be updated. If the measured period time has become stable then the two counters will start to count down and the HENA signal is set to LOW again, thereby disabling the register Hper and the SR flip-flop for Hpres parameter.

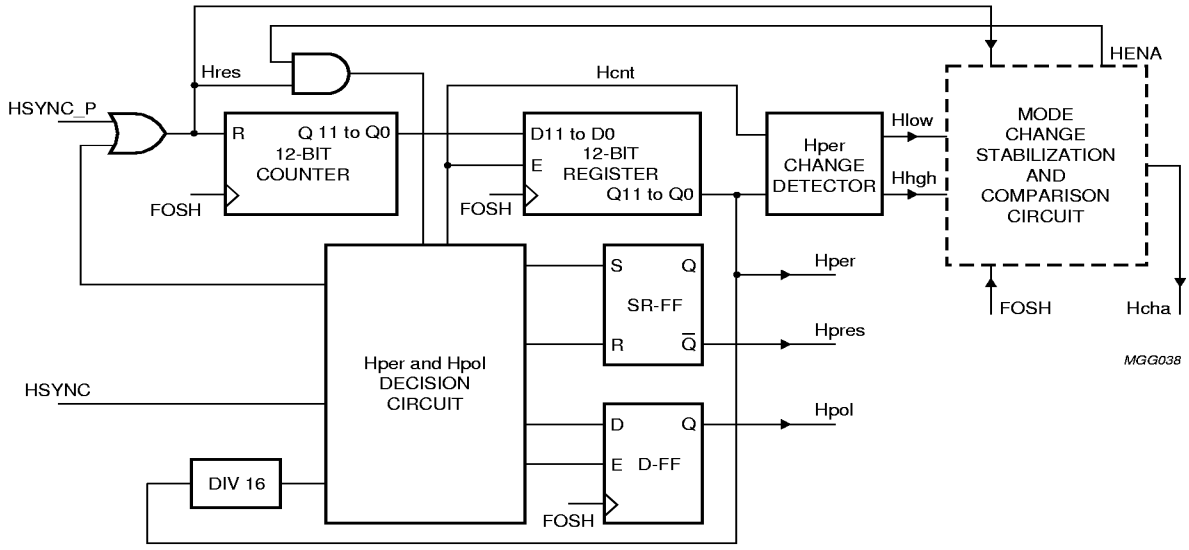


Fig.29 Horizontal mode detection.

19.2.6 VERTICAL MODE DETECTION

To detect the vertical mode parameters Vper, Vpol and Vpres, a circuit can be used which is almost similar to that for the horizontal mode detection. The only difference is that now the period time of one frame (instead of 4 lines) is measured. The presence indication for the vertical sync also contains a hysteresis of about 1.2%.

The formula to calculate the hysteresis is also similar to the condition of HSYNC processing. A 12-bit counter is used to count the clock number during one frame. Again FF0H (4080 decimal) and FC0H (4032 decimal) are taken as the threshold for Vfreq1 and Vfreq2 respectively, but the clock FOSV = FOSH/76. Vfreq1, Vfreq2 and Hysteresis are calculated as follows:

$$f_V = \text{Vfreq1} = \frac{\text{FOSV}}{(\text{Counter value})} = \frac{\text{FOSH}}{76 \times 4080}$$

$$f_V = \text{Vfreq2} = \frac{\text{FOSV}}{(\text{Counter value})} = \frac{\text{FOSH}}{76 \times 4032}$$

The hysteresis result,

$$\text{Hysteresis} = (\text{Vfreq1} - \text{Hfreq2}) / \text{Hfreq1},$$

is shown in Table 64.

If the new VSYNC frequency is larger than the previous VSYNC frequency, the time from mode change to interrupt issued is approximately from $((\text{Vperiod}_{(new)} \times 2) + \text{Vperiod}_{(previous)})$ to $(\text{Vperiod}_{(new)} \times 3)$. If the new VSYNC frequency is not in the static state and is smaller than the previous VSYNC frequency, the time from mode change to interrupt issued is approximately from $\text{Vperiod}_{(new)}$ to $\text{Vperiod}_{(new)} \times 2$. If the new VSYNC frequency is in the static state, the time from mode change to interrupt issued is approximately from $(4080 \times 6.3\ \text{us}) - \text{Vperiod}_{(previous)}$ to $4080 \times 6.3\ \mu\text{s}$.

Table 65 Hysteresis in the Vpres detection

(f_{clk}) (MHz)	FOSH (MHz)	Vfreq1 (MHz)	Vfreq2 (MHz)	HYSTERESIS (%)
10	10	32.25	32.63	1.2
12	12	38.70	39.16	1.2
16	8	25.80	26.11	1.2

The polarity can be measured by looking to the level of the input sync at 1/4 of the frame time. The mode change in vertical the frame period will be detected if there is a change for a longer time than 3 frame periods.

The accuracy of Vper can calculated by the following formula:

$$\text{Accuracy in V} = \frac{100\%}{(\text{vertical counter value})} = \frac{100\% \times f_V}{(\text{FOSV})} = \frac{100\% \times 76 \times f_V}{(\text{FOSH})}$$

The hysteresis results and the accuracy of Vper are shown in Table 63.

19.2.7 HORIZONTAL PULSE GENERATOR

Through the control flags, HPG1 and HPG0, the horizontal pulse generator is able to generate the HSYNC_{out} signal with the following formats:

1. The same pulse as the input horizontal sync, or a substitution pulse (with a fixed length) in case of a missing sync pulse.
2. A pulse starting at the beginning of the input horizontal sync but now with a fixed length, or a substitution pulse (with fixed length).
3. A free running sync pulse.
4. The same pulse as the input horizontal sync. The substitution pulse is inhibited even if HSYNC is missing.

To be able to generate a substitution pulse the down counter, depicted in Fig.23, is loaded at each incoming sync with a period time just a bit longer than the measured Hper. If now normal syncs are present then the counter will be loaded just before it reaches 000H. Therefore the counter will not generate a pulse (in Fig.23, HPST remains LOW). However, if a sync pulse is missing the counter will reach 000H and generates a substitution pulse, HPST. At the same time it will load itself again for the next run. To generate free running pulses the only thing to do is to load the down counter with the parameter HFP and to stop the load pulses coming from the input sync, HSYNC_P. So, the signal HPST is either a free running pulse or a pulse in case an input sync is missing.

The 5-bit down counter is used to generate a pulse with pulse width $t_{W(H)}$. As shown in Fig.23, the SR flip-flop will be set by either HPST (free running/missing sync pulse) or by the leading or trailing edge of the incoming sync pulse, HSYNC_P. In the same way the SR flip-flop will be reset by either the end of the incoming sync pulse or by the pulse width counter.

If necessary (HPG1, HPG0: 1, 1), the substitution pulse can be suppressed and only the incoming HSYNC pulse will be delivered out. The frequency of the free running pulse is: $f_{H(free)} = \frac{FOSH}{Hfp}$

The pulse width of the free running mode is:
$t_{W(H)} = T_{FOSH} \times (HOPW<4:0>)$

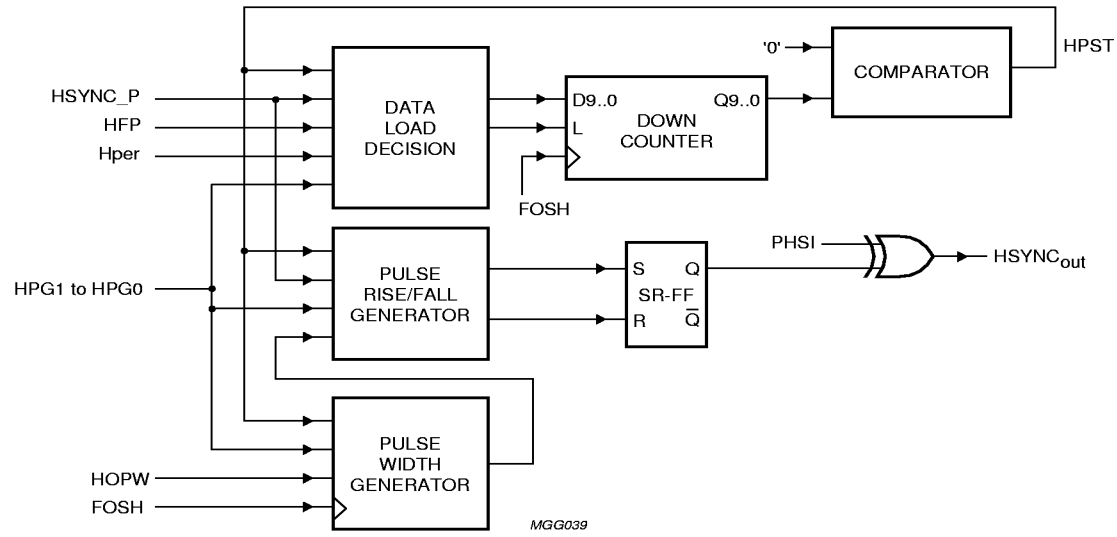


Fig.30 Horizontal pulse generator.

19.2.8 Vertical pulse generator

The clock used in this vertical pulse generator must be switchable between FOSV (needed to generate missing sync pulses) and HPST (free running syncs from the horizontal pulse generator) needed in the free running mode. In the latter situation one vertical sync pulse (i.e. one frame) will be generated by including an integer number of the horizontal scanning lines (i.e. HPST). In this case, the pulse width is also counted by HPST. The rest is more or less equal to the horizontal pulse generator.

The frequency of the free running pulse is:

$$f_{V(free)} = \frac{f_{H(free)}}{2 \times Vfp}$$

The pulse width of the free running mode is:
$t_{W(V)} = T_{H(free)} \times [(VOPW<3:0>) + 2] \times 2$

19.2.9 Capture registers

This function captures the 6 parameters, Hper, Hpol, Hpres, Vper, Vpol and Vpres such that the current value of these parameters is available for the microcontroller core (software) at all times. The mode change interrupt will be activated if:

- The horizontal period changes
- The vertical period changes
- The horizontal polarity changes
- The vertical polarity changes.

19.2.10 Control

All necessary control signals for the complete hardware mode detection are mentioned in Section 19.1.

19.2.11 The Display Pattern Generation

For certain events such as disconnection with the host or the life test of monitor sets, it is convenient to have certain display patterns shown on the monitor as the indication of the operation of the monitor. The P83C880 provides two simple patterns, the white pattern and the cross hatch pattern for this purpose in the free running mode.

In the free running mode, the intervals of HSYNC and VSYNC are determined by the 10-bit registers, HFP and VFP. Based on the HFP and VFP, the internal down counter will determine the starting or ending of the HSYNC and VSYNC. The starting position, ending position and the duration of the adjacent hatch lines are all related to the programmed value of HFP and VFP. As a matter of fact, the starting and ending position of the white pattern are decided by the fixed number of FOSH clock and FOSV clock (see Fig.31). Therefore, the position or dimension of the white pattern is much related to HFP and VFP. The duration of every two hatch lines (accordingly, the number of hatch lines in the horizontal or vertical direction) will depend on two fixed numbers (32, 32) which divide HFP and VFP. For both white and cross hatch patterns, the displayed pattern might look different in the different timing modes and the symmetric display is not guaranteed. However, they should be sufficient to be used as the indicator to report the status of the monitor. Figure 31 demonstrates the display of the two patterns.

Two flags: PATENA (SFR PWME2) and PATTYP (SFR PULCNT), are used to control the pattern display; for detailed usage of those control bits refer to Section 7.3.6 and Section 19.1.9.

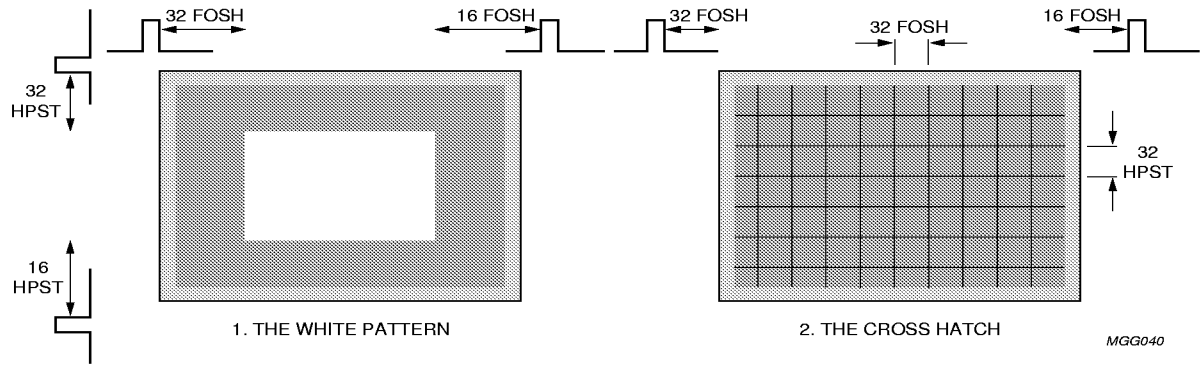


Fig.31 Two self test display patterns.

19.2.12 CLAMP output

To facilitate the video processing in the following stage, the clamping pulse can be delivered on pin PWM8/CLAMP/P3.0 by setting flag CLMPEN (PULCNT.7) to HIGH. The clamping pulse always accompanies the $HSYNC_{out}$ pulse. Therefore, even in the free running mode the clamping pulse is still present as long as the CLMPEN bit is set. Flag FBPO (PULCNT.1) can be used to choose the front porch clamp pulse (FBPO = 0) or the back porch clamp pulse (FBPO = 1). The pulse width of the clamping output signal is fixed to 8 FOSH clocks.

The bit PHSI (PULCNT.0) is used to set the output polarity of $HSYNC_{out}$ and the clamping pulse. If PHSI is LOW then the output polarity will be positive, otherwise the output polarity is negative.

19.3 System operation

After a successful power-on reset, the P83C880 will automatically start the mode detection and sync separation by default. The output pulses of HSYNC and VSYNC are delivered in the free running format by the default setting. Since it takes at least 5 frames to finalize the calculation of the HSYNC period, Hper, VSYNC period and Vper, it is better to wait until the stable state is reached to get the complete information after power-on reset. Accordingly, Hpol, Vpol, Hpres, Vpres and sync separation will be decided upon after Hper and Vper are settled. To prevent interfering the CPU too often, the interrupt request flag IE0 (SFR TCON) is only set if the mode (including frequency or polarity) is changed. IE0 will be cleared either by the CPU when the service routine is called (IT0 = 1) or by the service routine itself (IT0 = 0).

If it is required to reduce power dissipation or if it is preferred to stabilize the entire system after power-on reset before mode detection is proceeded, the bit MARCH (SFR MDCST) can be cleared to logic 0. Mode detection can be activated at the desired moment later on by setting MARCH to a logic 1.

In fact, the detection of the Device Power Management Signalling (DPMS) modes like 'Normal', 'Standby', 'Suspend', 'Power off', etc. and sync separation are mixed together with the display mode detection.

The mode detection function provides the hardware vehicle to facilitate the mode detection and related activities. Nevertheless, to use this facility to a maximum, the proper interaction between software and hardware is still essential. When VSYNC is absent and HSYNC is present, the polarity of HSYNC is always fixed. A software flow chart example is illustrated in Fig.32.

19.3.1 Display Power Management Signalling (DPMS) specification

According to the DPMS specification: Hfreq <10 kHz and Vfreq <10 Hz indicates that HSYNC and VSYNC are inactive. However, in the real application, there are no modes running with 10 kHz≤ Hfreq≤15 kHz and 10 Hz≤ Vfreq≤40 Hz. Therefore, Hfreq = 15 kHz and Vfreq = 40 Hz are chosen as the threshold to indicate an active or inactive signal for HSYNC and VSYNC respectively.

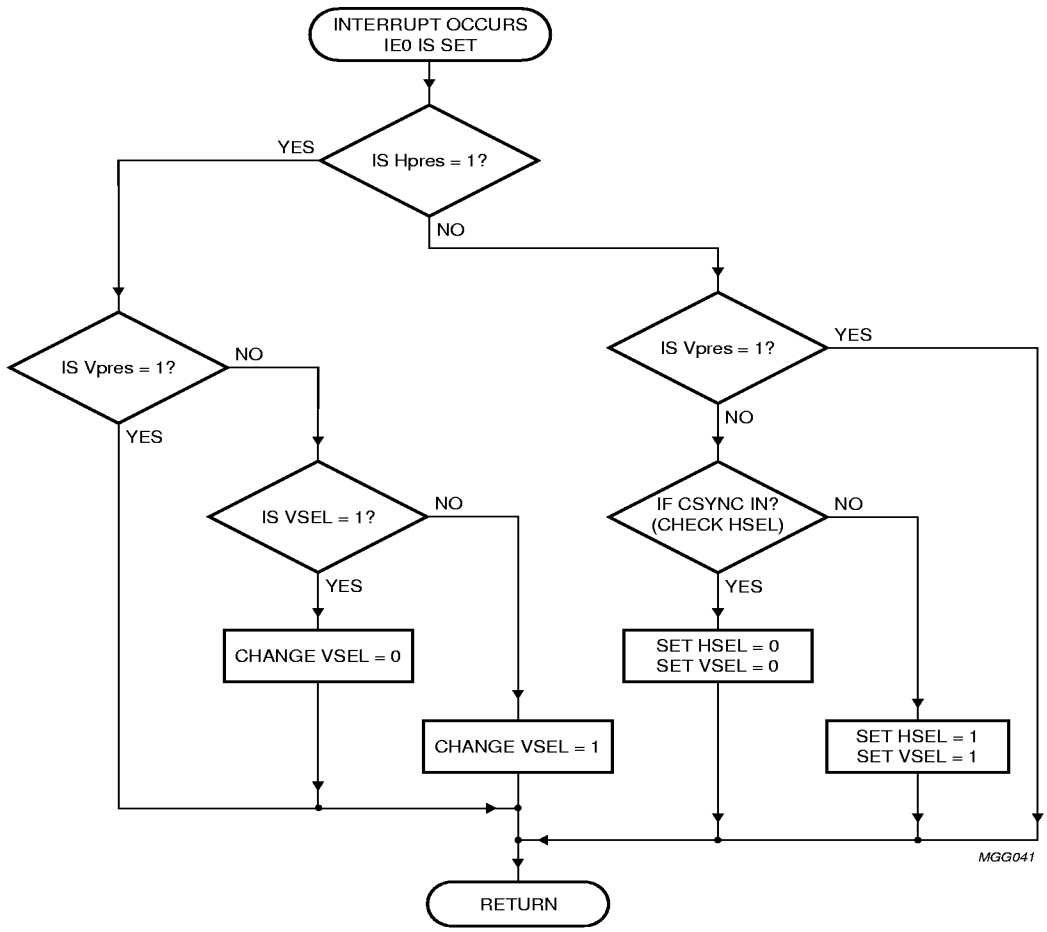


Fig.32 Software procedure in system operation: Mode detection flow chart.

20 POWER MANAGEMENT

The P83C880 supports power management. The hardware mode detection complies to the Device Power Management Signalling (DPMS) according to the specification of VESA. The microcontroller is able to distinguish between 'Normal', 'Standby', 'Suspend' and 'Power off' mode by detecting the frequency of HSYNC and VSYNC. HSYNC is an active signal if its frequency is above 15 kHz; otherwise, it is inactive. For VSYNC 40 Hz is the threshold to judge if it is active or inactive. See also Section 19.3.1. The exact values for the threshold depend on the chosen crystal oscillator frequency:
10, 12 or 16 MHz.

The information about the status of HSYNC and VSYNC can be represented by the bits, Hpres and Vpres in SFR MDCST (address F8H).

The combination of Hpres and Vpres in relation to the operating mode is shown in Table 66.

In the case of DDC2AB protocol, apart from hardware mode detection, the protocol of ACCESS.bus also supports Device Power Management Signalling (DPMS) commands. In some operating modes the microcontroller might enter a kind of power saving mode, i.e. Idle or Power-down mode. In this situation the extremely low retention voltage of 1.8 V for internal memory can support the microcontroller to save its state before switching to Idle or Power-down mode.

One set of I/O ports with the capability to drive LEDs can help to highlight the current operating mode of monitors. It is also possible to control the power consumption sources like heater, main supply etc., through an I/O port in a related operating mode.

Table 66 Display Power Management modes

OPERATING MODES	HSYNC	Hpres	VSYNC	Vpres
Normal	active	1	active	1
Standby	inactive	0	active	1
Suspend	active	1	inactive	0
Power off	inactive	0	inactive	0

21 CONTROL MODES

The microcontroller can operate in different control modes (e.g. for emulation or OTP programming) by means of a 10-bit serial code that is shifted into the RESET input. The signal at the XTAL1 pin serves as the shift clock.

The code is built up as follows: 0 XXXXX 0101; whereby the first '0' is the start bit, followed by a 5-bit code and completed by '0101'. The 5-bit code determines the mode.

Normal mode is reached when no code is shifted in, but the RESET pin remains HIGH for at least 10 cycles of the XTAL1 input plus 20 μs.

When an OTP Programming/Verification, Emulation or a Test mode should be entered, the timing must be as shown in Fig.34.

The RESET signal is sampled on the rising edge of XTAL1. Set-up and hold times for the RESET signal with respect to the XTAL1 rising edge are 10 ns each.

The procedure to enter a specific control mode is as follows:

1. Reset the microcontroller (CPU) by keeping the RESET signal HIGH for at least 10 XTAL1 clock cycles to escape any other control mode, plus at least 20 μs (minimum of 24 internal clock cycles) to reset the internal logic.
2. Shift in the specific code for the required control mode (e.g. 0 1 1000 0101 for Emulation mode) via the RESET pin and the XTAL1 clock (MSB first).
3. After the 10th bit the RESET signal should go LOW within 9 cycles of XTAL1, otherwise the microcontroller will enter Normal mode.

For the OTP Programming/Verification control modes see Chapter 22.

22 ONE TIME PROGRAMMABLE (OTP) VERSION

The OTP version P87C380 contains:

- 32 kbytes ROM
- 512 bytes RAM.

Next to the 32 kbytes ROM, another extra 384 bytes are available. These bytes can be used for identification purposes or to indicate the version number, etc. These bytes are addressed via address lines A0 to A4.

Access to the OTP for program execution is done in Normal mode. For writing and reading (Verification) of the program memory and of the extra 384 bytes, different control modes are used as shown in Table 67.

The signals and waveforms are given in Table 69 and Fig.33. Pin connections to be used during Programming and Verification are given in Table 68.

Table 67 Code for Programming/Verification of the OTP version P87C180A

MEMORY	CODE
ROM: 32 kbytes (program memory)	0100000101
Extra 384 bytes (memory)	0100010101
ROM: 32 kbytes (verification)	0111110101

Table 68 Pin assignments during Programming/Verification mode

PIN	CONNECT TO
1	V_{SS}
2	V_{SS}
3	A13
4	A12
5	A11
6	A10
7	A9
8	A8
9	XTAL1 (note 1)
10	XTAL2 (note 1)
11	V_{DD}
12	V_{SS}
13	$\overline{WE}$
14	DI5/DO5
15	DI6/DO6
16	$\overline{OE}$
17	DI4/DO4
18	A7
19	A6
20	A5
21	A4
22	A3
23	A2
24	A1
25	A0
26	RESET
27	V_{SS}
28	V_{SS}
29	V_{SS}
30	V_{SS}
31	V_{SS}
32	V_{DD}
33	V_{SS}
34	A14
35	V_{PP}
36	DI3/DO3
37	DI2/DO2
38	DI1/DO1
39	DI0/DO0
40	DI7/DO7
41	V_{SS}
42	V_{SS}

Note

1. For driving the oscillator pins see Fig.10c.

Table 69 Timing table for programming; see Fig.33

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
$t_{su(AW)}$	address set-up time write	2	–	–	µs
$t_{h(AW)}$	address hold time write	10	–	–	µs
$t_{su(AR)}$	address set-up time read	10	–	–	ns
$t_{h(AR)}$	address hold time read	200	–	–	ns
$t_{su(D)}$	data set-up time	2	–	–	µs
$t_{h(D)}$	data hold time	20	–	–	µs
$t_{su(prog)}$	programming voltage set-up time	10	–	–	µs
$t_{h(prog)}$	programming voltage hold time	10	–	–	µs
$t_{W(prog)}$	programming pulse width	90	–	–	µs
$t_{W(R)}$	read pulse width	300	–	–	ns
$t_{ACC(OE)}$	output enable access verify	92	122	183	ns
$t_{DZ(OE)}$	data float after $\overline{OE}$	10	–	–	ns

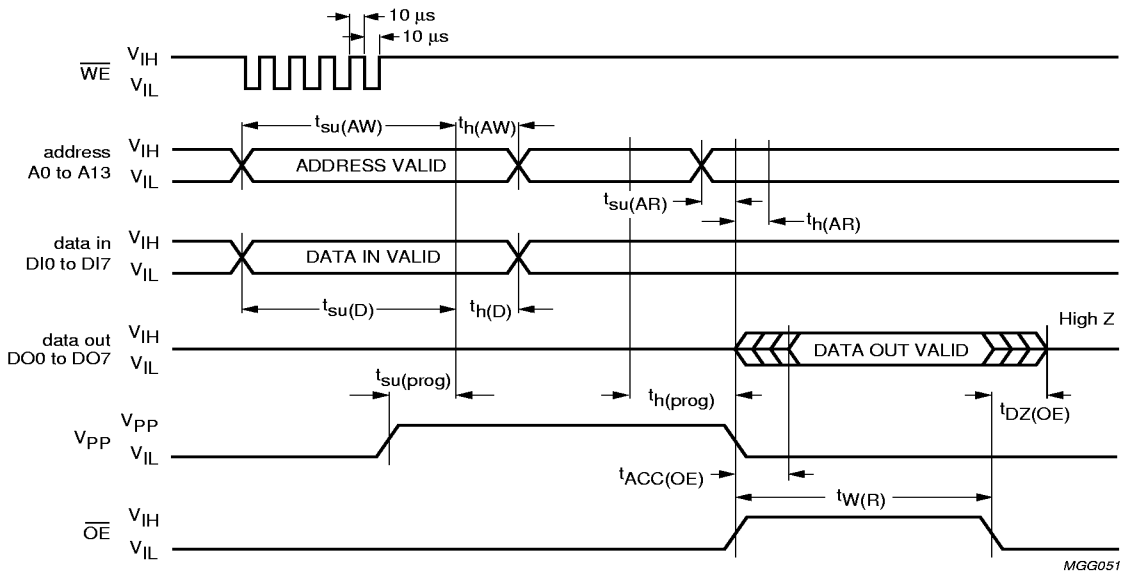


Fig.33 Programming waveforms.

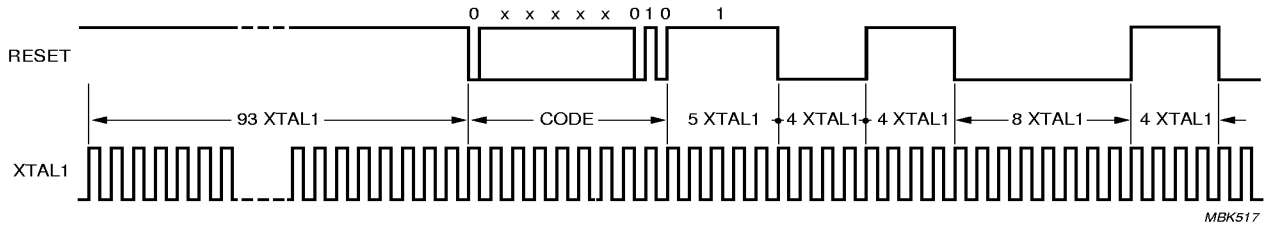


Fig.34 Timing for OTP Programming/Verification.

23 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	supply voltage	4.4	5.5	V
V_I	input voltage (all inputs)	−0.5	V_{DD} + 0.5	V
$I_{source(max)}$	total maximum source current for all port lines	0.50	0.67	mA
$I_{sink(max)}$	total maximum sink current for all port lines	155	165	mA
P_{tot}	total power dissipation	88.4	170	mW
T_{stg}	storage temperature	−60	+150	°C
T_{amb}	operating ambient temperature (for all devices)	−25	+85	°C

24 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

25 DC CHARACTERISTICS

V_{DD} = 4.5 to 5.5 V; V_{SS} = 0 V; T_{amb} = −25 to +85 °C; all voltages with respect to V_{SS} unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	operating supply voltage		4.4	5	5.5	V
I_{DD}	operating supply current	f_{clk} = 12 MHz	20.1	29.4	30.9	mA
		f_{clk} = 16 MHz	25.2	36.1	38.4	mA
V_{POR}	Power-on-reset voltage level		3.6	3.9	4.2	V
PROGRAMMING SUPPLY						
$V_{DD(P)}$	supply voltage programming mode		4.4	5.0	5.5	V
V_{PP}	programming voltage		12.0	12.75	13.0	V
$I_{DD(P)}$	supply current programming mode		10.6	19.1	24.0	mA
I_{PP}	programming current		77.8	80.1	81.3	mA
RESET						
$R_{I(RESET)}$	input resistance RESET	V_{DD} = 4.5 V to 5.5 V	20	60	180	kΩ
I_{LI}	input leakage current	$V_{SS} < V_I < V_{DD}$			±10	µA
V_{IL}	LOW-level input voltage		V_{SS} − 0.5	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	V_{DD} + 0.5	V
HSYNC$_{in}$/$\overline{\text{PROG}}$ and VSYNC$_{in}$/$\overline{\text{OE}}$						
I_{IL}	Input leakage current	$V_{SS} < V_I < V_{DD}$			−10	µA
V_{IL}	LOW-level input voltage		V_{SS} − 0.5	–	0.8	V
V_{IH}	HIGH-level input voltage		2.0	–	V_{DD} + 0.5	V
CSYNC$_{in}$/P1.6						
I_{LI}	input leakage current	$V_{SS} < V_I < V_{DD}$			−10	µA
V_{IL}	LOW-level input voltage		V_{SS} − 0.5	–	0.8	V
V_{IH}	HIGH-level input voltage		2.0	–	V_{DD} + 0.5	V
I_{OL}	LOW-level output sink current	$V_O \leq 0.4$ V	1.6	–	–	mA
		$V_O \leq 1.0$ V	10	–	–	mA
I_{OH}	HIGH-level pull-up output source current					
	strong pull-up during 2 clock cycles	$V_O = V_{DD} - 0.4$ V	1.6	–	–	mA
	weak pull-up	$V_O = V_{DD} - 0.4$ V	25	–	–	µA
Port 0						
V_{IL}	LOW-level input voltage		V_{SS} − 0.5	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	V_{DD} + 0.5	V
I_{LI}	input leakage current	$V_I = 0.4V_{DD}$; V_{DD} = 5 V	−10	–	−100	µA
I_{IT}	input transition current	$V_I = 0.5V_{DD}$; V_{DD} = 5 V	–	–	−1000	µA
I_{OL}	LOW-level output sink current	$V_O \leq 0.4$ V	1.6	–	–	mA
		$V_O \leq 1.0$ V	10	–	–	mA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{OH}	HIGH-level pull-up output source current					
	strong pull-up during 2 clock cycles	$V_O = V_{DD} - 0.4$ V	1.6	–	–	mA
	weak pull-up	$V_O = V_{DD} - 0.4$ V	25	–	–	µA
Port 1: SCL/P1.0, SDA/P1.1, SCL1/P1.2 and SDA1/P1.3						
V_{IL}	LOW-level input voltage		$V_{SS} - 0.5$	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	$V_{DD} + 0.5$	V
I_{LI}	input leakage current	$V_I = 0.4V_{DD}$; $V_{DD} = 5$ V	–10	–	–100	µA
I_{IT}	input transition current	$V_I = 0.5V_{DD}$; $V_{DD} = 5$ V	–	–	–1000	µA
I_{OL}	LOW-level output sink current	$V_O = 0.4$ V; $V_{DD} = 5$V	3.0	–	–	mA
I_{OH}	HIGH-level pull-up output source current					
	strong pull-up during 2 clock cycles	$V_O = V_{DD} - 0.4$ V	1.6	–	–	mA
	weak pull-up	$V_O = V_{DD} - 0.4$ V	25	–	–	µA
Port 1 and 3: $VSYNC_{out}$/P1.4, $HSYNC_{out}$/P1.5, PWM10/P1.7 and ADC0/P3.2 to ADC1/P3.3						
V_{IL}	LOW-level input voltage		$V_{SS} - 0.5$	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	$V_{DD} + 0.5$	V
I_{LI}	input leakage current	$V_I = 0.4V_{DD}$; $V_{DD} = 5$ V	–10	–	–100	µA
I_{IT}	input transition current	$V_I = 0.5V_{DD}$; $V_{DD} = 5$ V	–	–	–1000	µA
I_{OL}	LOW-level output sink current	$V_O \leq 0.4$ V	1.6	–	–	mA
I_{OH}	HIGH-level pull-up output source current					
	strong pull-up during 2 clock cycles	$V_O = V_{DD} - 0.4$ V	1.6	–	–	mA
	weak pull-up	$V_O = V_{DD} - 0.4$ V	25	–	–	µA
Port 2 and 3: P2.0/PWM0 to P2.7/PWM7 and Port P3.0/PWM8/CLAMP to P3.1/PWM9/PATOUT						
V_{IL}	LOW-level input voltage		$V_{SS} - 0.5$	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	$V_{DD} + 0.5$	V
I_{LI}	input leakage current	$V_I = 0.4V_{DD}$; $V_{DD} = 5$ V	–10	–	–100	µA
I_{IT}	input transition current	$V_I = 0.5V_{DD}$; $V_{DD} = 5$ V	–	–	–1000	µA
I_{OL}	LOW-level output sink current	$V_O \leq 0.4$ V	1.6	–	–	mA
I_{OH}	HIGH-level pull-up output source current					
	strong pull-up during 2 clock cycles	$V_O = V_{DD} - 0.4$ V	1.6	–	–	mA
	weak pull-up	$V_O = V_{DD} - 0.4$ V	25	–	–	µA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$\overline{INT1}/V_{PP}$						
V_{IL}	LOW-level input voltage		$V_{SS} - 0.5$	–	$0.2V_{DD} - 0.1$	V
V_{IH}	HIGH-level input voltage	I_{IH} = 2 mA	$0.2V_{DD} + 0.9$	–	12.6	V
$V_{IH} - V_{DD}$	input voltage with respect to V_{DD}		–	–	8	V

Note

1. This maximum applies at all times, including during power switching, and must be accounted for in power supply design. During a power-on process, the +12 V source used for external pull-up resistors should not precede the V_{DD} of the P83C880, P83C180, P83C280 and P83C380 up their respective voltage ramps by more than this margin, nor, during a power-down process, should V_{DD} precede +12 V down their respective voltage ramps by more than this margin.

26 DIGITAL-TO-ANALOG CONVERTER CHARACTERISTICS

V_{DD} = 5 V; V_{DDA} = 5 V; T_{amb} = 27 °C.

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{DD}	supply voltage		4.4	–	5.5	V
RES_{DAC}	DAC resolution		–	8	–	bit
$I_{o(DAC)}$	operating current for each DAC channel	output pin connected to V_{DDA} via 40 kΩ	–	–	2.5	mA
$I_{o(DAC)(id)}$	DAC idle current	all 4 DAC channels shut-down	–	–	61	μA
$V_{o(DAC)}$	DAC output level voltage	output pin open	0	–	V_{DDA}	V
		at FFH code input; I_{OH} = –1mA (note 1)	$V_{DDA} - 0.2$	–	–	V
		at 00H code input; I_{OL} = 1mA	–	–	0.2	V
DL_e	differential non-linearity	output level 0.2 to 4.8 V	–1	–	+1	LSB
t_{st}	settling time		–	–	20	ms
$\Delta V_{o(T)}$	output voltage variation due to temperature drift		–	–	1.5	mV/°C
ΔV_o	output voltage ripple	T_{test} > 20 ms	–	–	2	mV

Note

1. The ‘negative’ value denotes that the current is sourcing out from the chip into the load device through the output pin.

27 AC CHARACTERISTICS

V_{DD} = 4.5 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −25 to + 85 °C; all voltages with respect to V_{SS} unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
f_{clk}	crystal oscillator frequency	V_{DD} = 5 V	10	12	16	MHz
f_{PXE}	PXE resonator frequency	V_{DD} = 5 V	10	12	16	MHz
C_{XTAL1}	external capacitance at XTAL1					
	with crystal resonator		–	20	27	pF
	with PXE resonator		–	20	27	pF
C_{XTAL2}	external capacitance at XTAL2					
	with crystal resonator		–	20	27	pF
	with PXE resonator		–	20	27	pF
Analog-to-Digital Converter (software)						
$V_{in(A)}$	comparator analog input voltages ADC0; ADC1		V_{SS}	–	V_{DD}	V
V_{AE}	conversion error range		–	–	±½	LSB
$t_{conv(ADC)}$	conversion time (from any change in ADC input i.e. voltage level)		–	–	7	µs
DDC1 Mode[1] **(transmit-only, unidirectional); see Figs** 25 **and** 35						
t_{DOV}	output valid from VCLK		–	–	30	µs
t_{VCLKH}	VCLK high time		20	–	–	µs
t_{VCLKL}	VCLK low time		20	–	–	µs
$t_{t(mode)}$	mode transition time		–	–	800	ns
$t_{sup(input)}$	input filter spike suppression; for SDA, SCL and VSYNC		–	–	200	ns
$t_{su(DDC1)}$	DDC1 mode set-up time		–	5	–	µs
Horizontal sync input for the mode detection; see Fig.36						
Hfreq	HSYNC input frequency		15	–	150	kHz
$t_{W(HSYNC)}$	HSYNC input pulse width	notes 2 and 3	0.25	–	8	µs
H_{DCYC}	HSYNC input duty cycle	notes 2 and 3	–	–	25	%
$t_{ir(HSYNC)}$	HSYNC input rise time		–	–	100	ns
$t_{if(HSYNC)}$	HSYNC input fall time		–	–	100	ns
Vertical Sync Input for the Mode Detection; see Fig.37						
Vfreq	VSYNC input frequency		40	–	200	Hz
$t_{W(VSYNC)}$	VSYNC input pulse width	note 4	1	–	24	note 5
V_{DCYC}	VSYNC input duty cycle	note 4	–	–	25	%
$t_{ir(VSYNC)}$	VSYNC input rise time		–	–	100	ns
$t_{if(VSYNC)}$	VSYNC input fall time		–	–	100	ns

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Composite Sync Input for the Mode Detection; see Figs 38 and 39						
$T_{EQ(CSYNC)}$	equalizing pulse period for CSYNC input		–	0.5	–	note 5
$t_{WEQ(CSYNC)}$	equalizing pulse width for CSYNC input		–	0.5	–	note 6
NEQ	equalizing pulses plus the VSYNC interval		–	–	30	note 5
Horizontal and Vertical Pulse Generation; see Figs 40 and 41						
$T_{H(free)}$	horizontal pulse period in free running		10	–	66	µs
$t_{WH(free)}$	horizontal pulse width in free running		0.4	–	2.5	µs
$t_{or(HSYNC)}$	HSYNC output pulse rise time		30	31	32	ns
$t_{of(HSYNC)}$	HSYNC output pulse fall time		24	26	28	ns
$t_{d(in-out)(HSYNC)}$	HSYNC output delay to input		–	–	100	ns
$t_{dmax(HSYNCm)}$	HSYNC output maximum delay after missing HSYNC		–	–	350	ns
$t_{W(HSYNCsub)}$	HSYNC output substitution pulse width		–	–	2.5	µs
$t_{d(HSYNC-CLAMP)}$	CLAMP pulse delay to HSYNC input		0	–	125	ns
$t_{W(CLAMP)}$	CLAMP pulse width		0.4	–	2.5	µs
$T_{V(free)}$	vertical pulse period in free running		–	–	2048	note 5
$t_{WV(free)}$	vertical pulse width in free running		–	–	32	note 5
$t_{or(VSYNC)}$	VSYNC output pulse rise time		22	24	26	ns
$t_{of(VSYNC)}$	VSYNC output pulse fall time		24	27	30	ns
$t_{d(in-out)(VSYNC)}$	VSYNC output delay to input		–	–	250	ns
$t_{dmax(VSYNCm)}$	VSYNC output maximum delay after missing VSYNC		–	–	1	note 5
$t_{W(VSYNCsub)}$	VSYNC output substitution pulse width		–	–	16	note 5
Slope control port						
$t_{tLH(sce)}$	LOW-to-HIGH transition delay while slope control enabled	$0.1V_{DD}$ to $0.9V_{DD}$; note 7	25	–	100	ns
$t_{tHL(sce)}$	HIGH-to-LOW transition delay while slope control enabled	$0.9V_{DD}$ to $0.1V_{DD}$; note 7	25	–	100	ns
$t_{PLH(scd)}$	LOW-to-HIGH propagation delay while slope control disabled	$0.1V_{DD}$ to $0.9V_{DD}$; load = 50 pF	–	–	10	ns
$t_{PHL(scd)}$	HIGH-to-LOW propagation delay while slope control disabled	$0.9V_{DD}$ to $0.1V_{DD}$; load = 50 pF	–	–	10	ns

Notes

1. For the DDC2 Mode (bidirectional I^2C-bus mode), refer to the standard I^2C-bus specification.
2. The actual sync width has to fulfil both requirements, maximum pulse width $t_{W(HSYNC)}$ = 8.0 µs and maximum duty cycle H_{DCYC} of 25%, at the same time.
3. The polarity of the horizontal sync pulse is said to be positive if the HIGH period is shorter than the LOW period.
4. The actual sync width has to fulfil both requirements, maximum pulse width $t_{W(VSYNC)}$ = 24 horizontal line periods and maximum duty cycle V_{DCYC} of 25%, at the same time.

5. Unit is HSYNC line period: $T_{line(HSYNC)}$.
6. Unit is HSYNC pulse width: $t_{W(HSYNC)}$.
7. This transition delay has to be loaded independent for capacitances up to 50 pF.

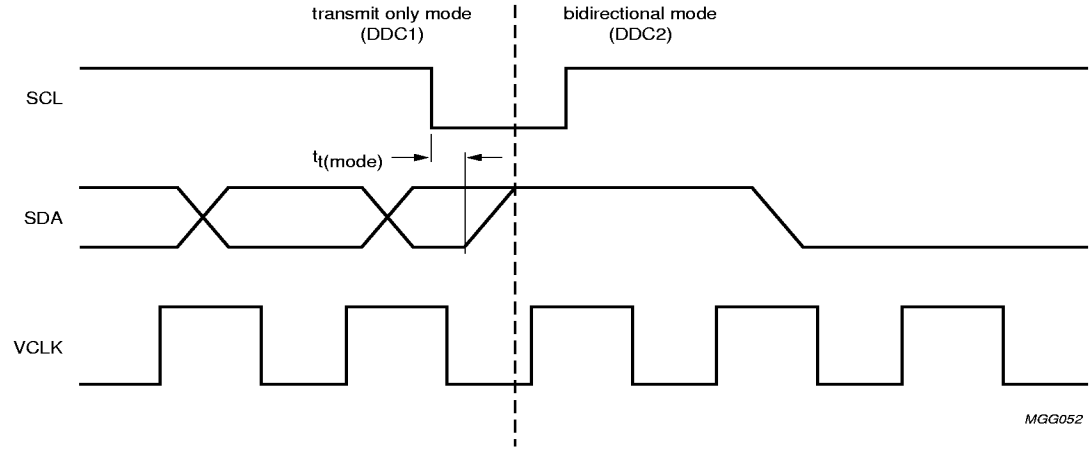


Fig.35 Mode transition from DDC1 to DDC2.

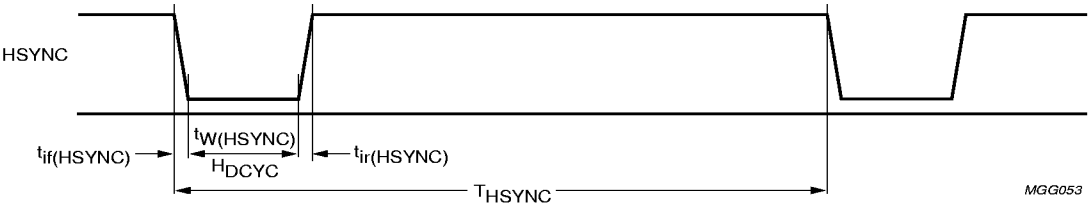


Fig.36 Horizontal sync input signal.

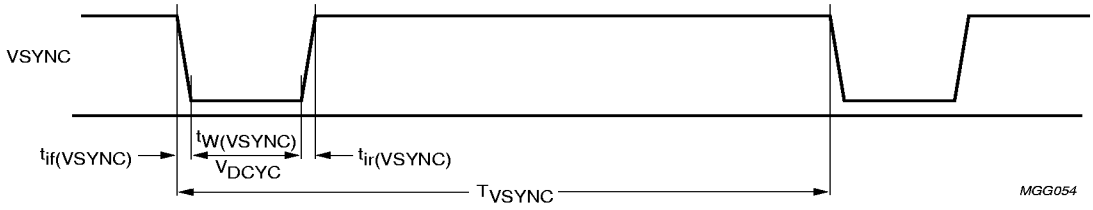


Fig.37 Vertical sync input signal.

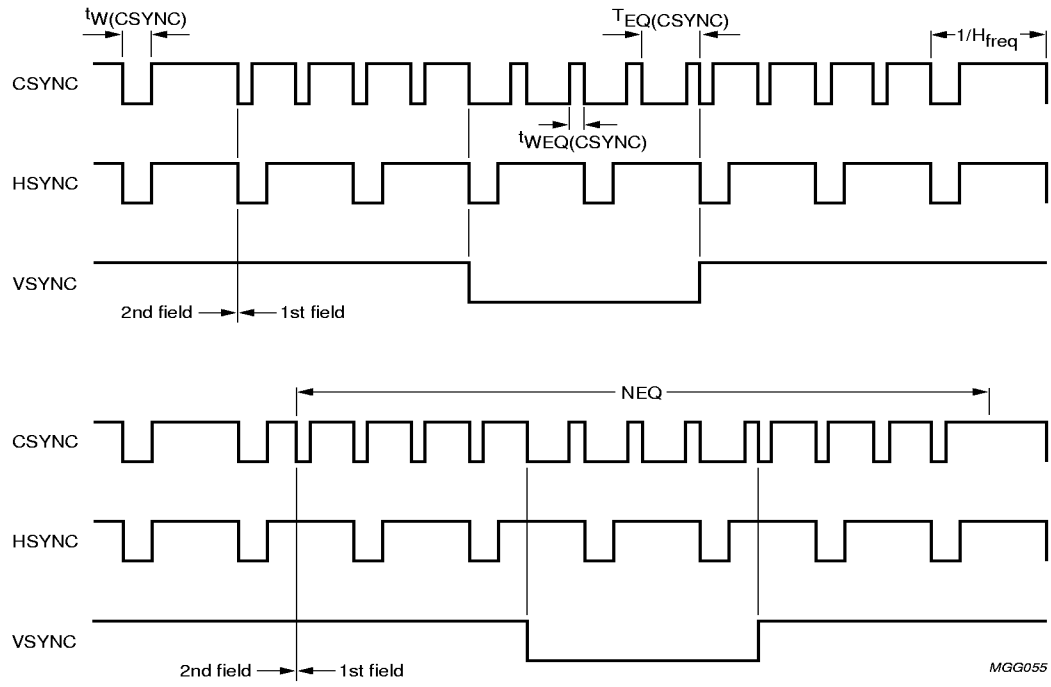


Fig.38 Standard composite sync pulse wave forms.

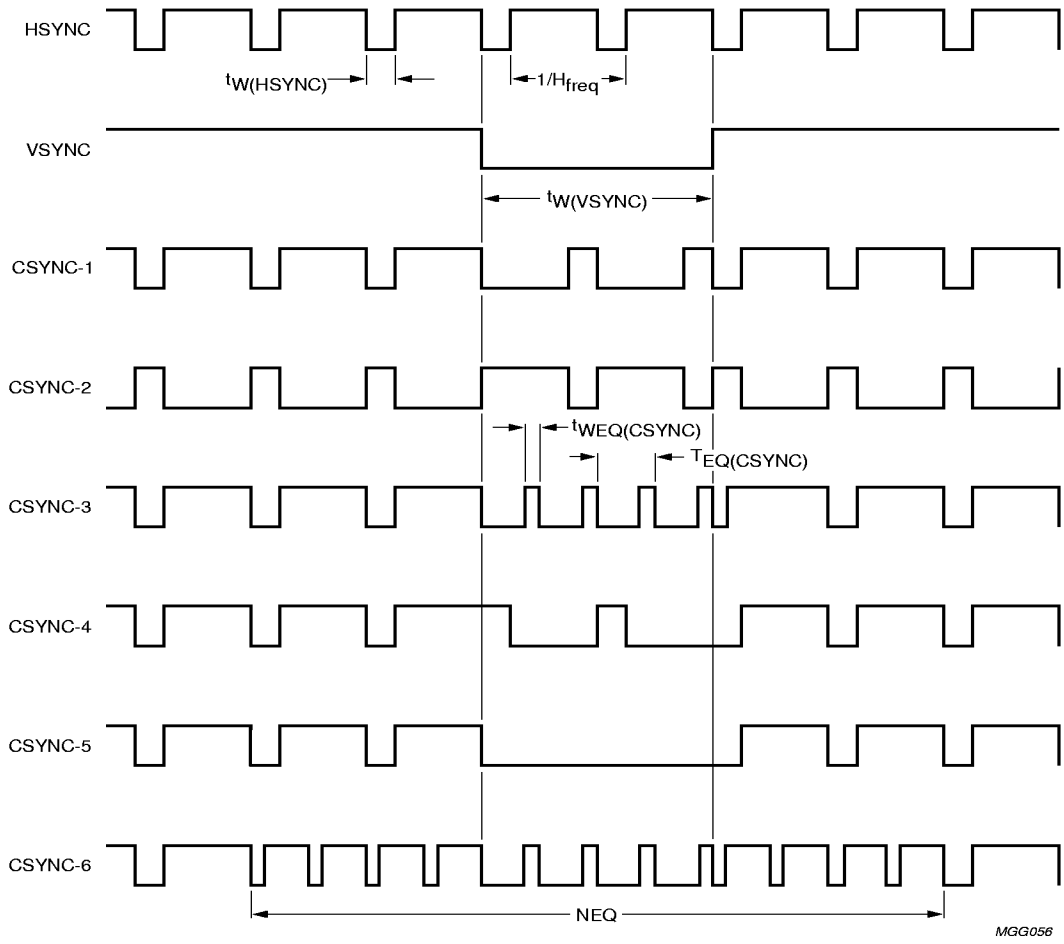


Fig.39 Composite sync pulse wave forms.

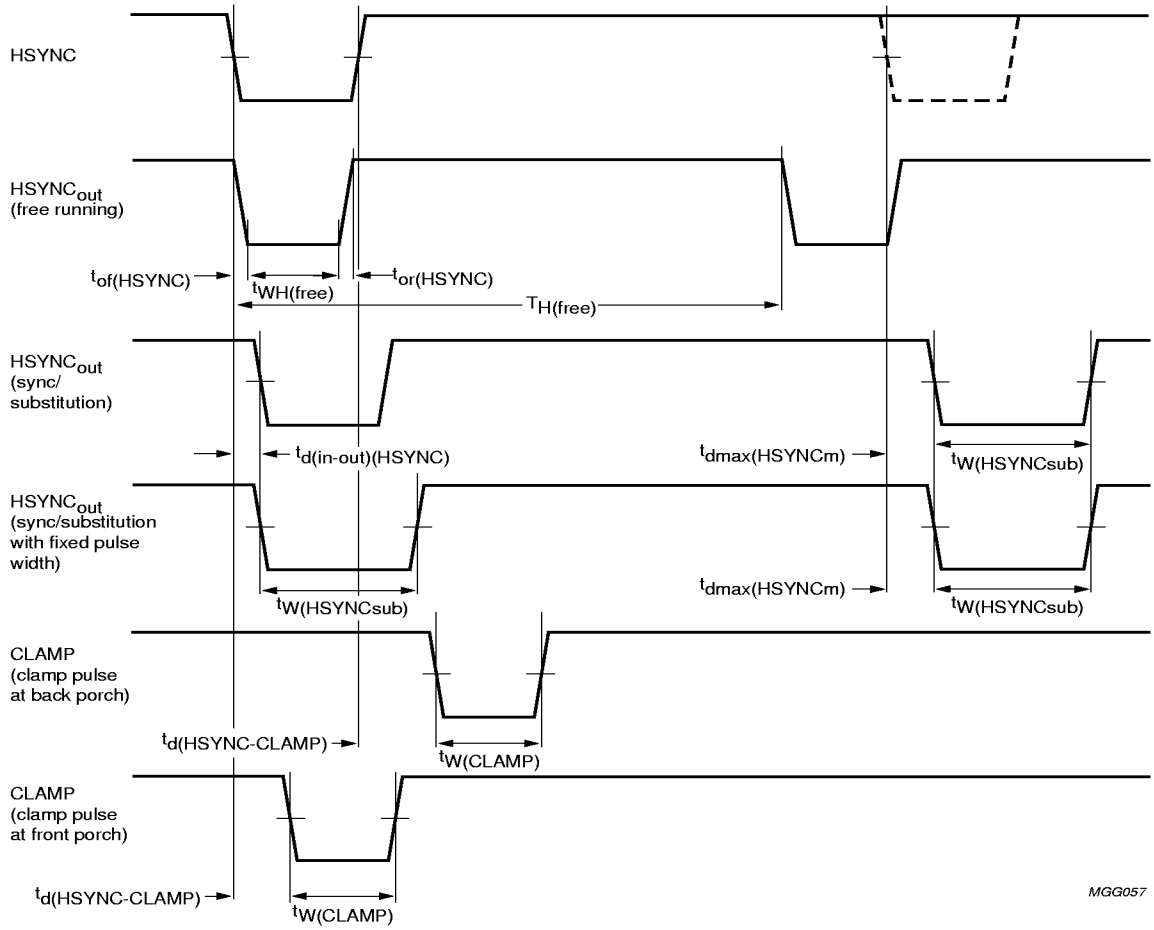


Fig.40 Horizontal sync output pulse.

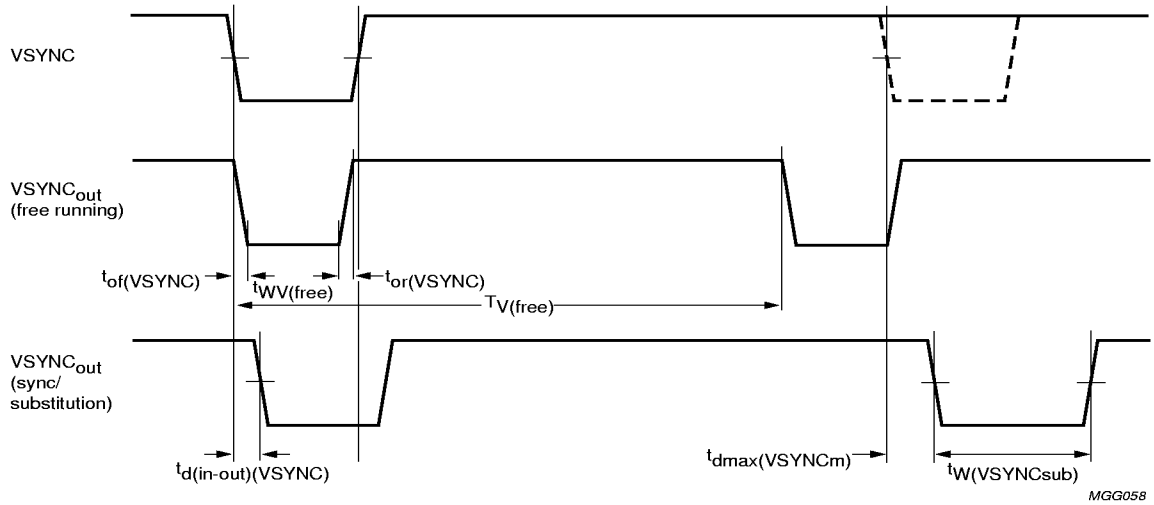
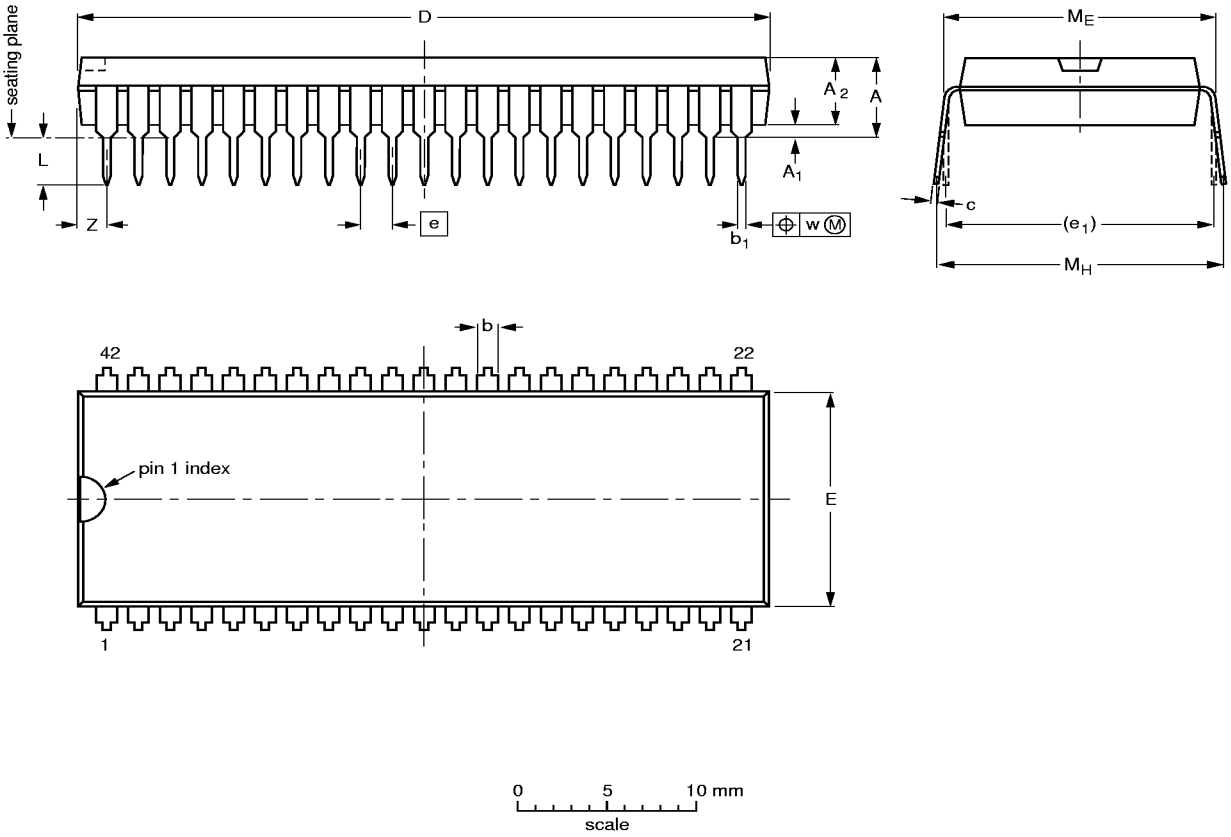


Fig.41 Vertical sync output pulse.

28 PACKAGE OUTLINE

SDIP42: plastic shrink dual in-line package; 42 leads (600 mil) **SOT270-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	5.08	0.51	4.0	1.3 0.8	0.53 0.40	0.32 0.23	38.9 38.4	14.0 13.7	1.778	15.24	3.2 2.9	15.80 15.24	17.15 15.90	0.18	1.73

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT270-1						~~90-02-13~~ 95-02-04

29 SOLDERING

29.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"IC Package Databook"* (order code 9398 652 90011).

29.2 Soldering by dipping or by wave

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg\ max}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

29.3 Repairing soldered joints

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.