DATA SHEET *REV. 2.0 (PRELIMINARY)*

HMS9xC7132

Microcontroller for monitor
with DDC, I2C, PWM, ADC and Sync. processor

SEP. 5, 2000

CONTENTS

1. GENERAL DESCRIPTION

2. FEATURES

3. BLOCK DIAGRAM

4. PINNING INFORMATION

5. MEMORY ORGANIZATION

6. INTERRUPT SYSTEM

7. POWER-SAVING MODES

8. INPUT / OUTPUT (I/O)

9. OSCILLATOR

10. RESET

11. WATCHDOG TIMER

12. TIMER

13. DDC INTERFACE

14. I2C INTERFACE

15. PULSE WIDTH MODULATION
  15.1 STATIC PWM
  15.2 DYNAMIC PWM

16. SYNC PROCESSOR

17. ANALOG-TO-DIGITAL CONVERTER

18. OPERATION MODE
  18.1 OTP MODE
  18.2 EMULATION MODE

19. DC CHARACTERISTICS

20. AC CHARACTERISTICS

21. PACKAGE DIMENSIONS

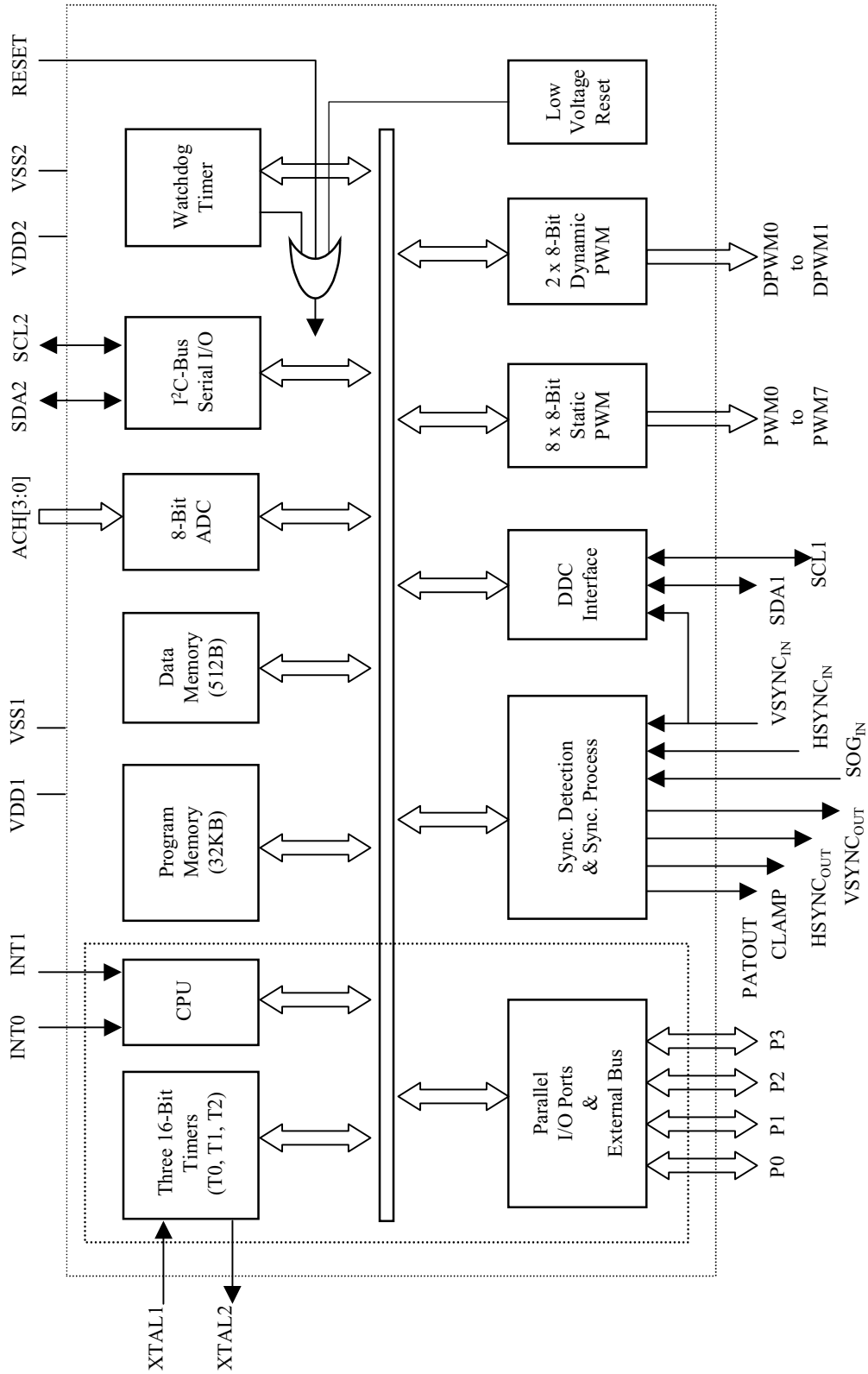
1. GENERAL DESCRIPTION

The HMS9xC7132 is a single-chip microcontroller of the 80C51 family, which is dedicated for monitor application. It is particularly suitable for multi-sync computer monitor controller. This contains DDC interfaces to the PC host, sync-detector and sync-processor for auto-sync application, ADC, static PWM, dynamic PWM and I2C bus interface for control of the video and deflection functions of the monitor.

2. FEATURES

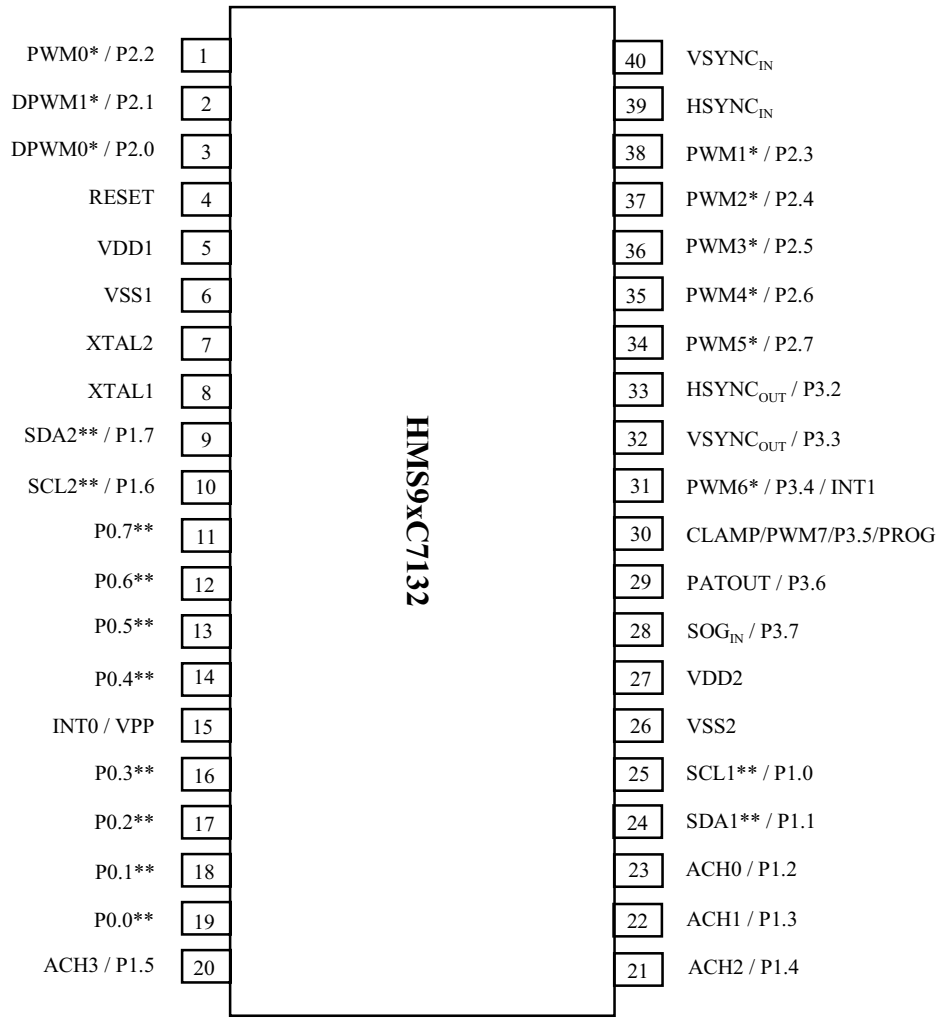
- 80C51 core
- 32K bytes of ROM for HMS91C7132
  (32K bytes of OTP ROM for HMS97C7132)
- 256 bytes of RAM and 256 bytes of AUX-RAM
- Three 16-bit timer/counters (T0, T1, T2)
- One DDC compliant interface :
  - Fully supports DDC1 with dedicated hardware
  - DDC2B, DDC2AB and DDC2B+ compliant dedicated hardware based on an I2C bus interface
  - RAM buffer with programmable size, 128 bytes or 256 bytes, which can be used for DDC operation or shared as system RAM
- On-chip sync processor
  - HSYNC frequency with 12-bit resolution
  - VSYNC frequency with 12-bit resolution
  - HSYNC and VSYNC polarity
  - HSYNC and VSYNC presence
  - Composite sync separation
  - Free running mode
  - Clamping
  - Pattern generation
  - Missing insertion option
- One 100K/400K bit/s hardware I2C bus interface for control of other system IC's
- Eight 8-bit Static PWM outputs for digital control applications
- Two 8-bit Dynamic PWM outputs for various waveform generation
- One 8-bit ADC with 4 input channels
- LED driver port ; two port lines with 15 mA drive capability
- One 8-bit port only for I/O function
- 24 derivative I/O ports configurable for alternative functions
- Watchdog timer (reset period : 524ms)
- On-chip low VDD voltage reset
- Two external interrupts available
- Special idle and power-down modes with low power consumption
- Operating frequency : 12 MHz
- Operating temperature : 0°C to 70°C
- Power supply : 4.5V to 5.5V

3. BLOCK DIAGRAM



4. PINNING INFORMATION

4.1 40 PDIP



* : Open-drain option
** : Open-drain type pin

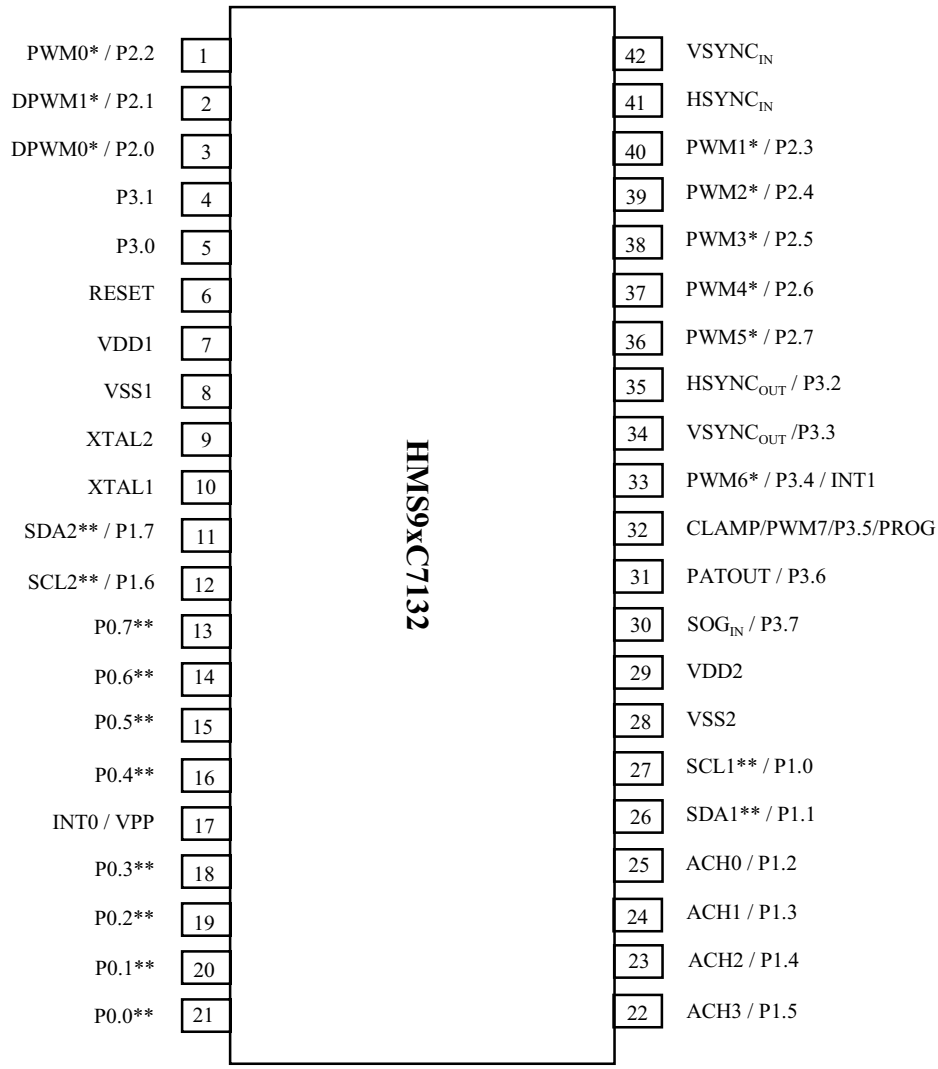
Fig. 4.1.1 40 PDIP Pinning

Table 4.1 40 PDIP Pin Descriptions

SYMBOL	PIN	DESCRIPTION
PWM0 / P2.2	1	8-bit Pulse Width Modulation output0 ; General I/O port P2.2
DPWM1 / P2.1	2	8-bit Dynamic Pulse Width Modulation output0 ; General I/O port P2.1
DPWM0 / P2.0	3	8-bit Dynamic Pulse Width Modulation output1 ; General I/O port P2.0
RESET	4	Reset input
VDD1	5	Power supply1 (+5V)
VSS1	6	Ground1
XTAL2	7	Oscillator output pin for system clock
XTAL1	8	Oscillator input pin for system clock
SDA2 / P1.7	9	I2C serial data I/O port ; General I/O port P1.7
SCL2 / P1.6	10	I2C serial clock I/O port ; General I/O port P1.6
P0.7	11	General I/O port P0.7 ; adapted for LED driver
P0.6	12	General I/O port P0.6 ; adapted for LED driver
P0.5	13	General I/O port P0.5
P0.4	14	General I/O port P0.4
INT0 / VPP	15	External interrupt input0 ; Programming supply voltage (during OTP programming)
P0.3	16	General I/O port P0.3
P0.2	17	General I/O port P0.2
P0.1	18	General I/O port P0.1
P0.0	19	General I/O port P0.0
ACH3 / P1.5	20	ADC channel3 input ; General I/O port P1.5
ACH2 / P1.4	21	ADC channel2 input ; General I/O port P1.4
ACH1 / P1.3	22	ADC channel1 input ; General I/O port P1.3
ACH0 / P1.2	23	ADC channel0 input ; General I/O port P1.2
SDA1 / P1.1	24	I2C serial data I/O port for the DDC interface ; General I/O port P1.1
SCL1 / P1.0	25	I2C serial clock I/O port for the DDC interface ; General I/O port P1.0

SYMBOL	PIN	DESCRIPTION
VSS2	26	Ground2
VDD2	27	Power supply2 (+5V)
SOGin / P3.7	28	Sync on Green input ; General I/O port P3.7
PATOUT / P3.6	29	Pattern out ; General I/O port P3.6
CLAMP / PWM7 / / P3.5 / PROG	30	Clamp out ; 8-bit Pulse Width Modulation output7 ; General output only port P3.5 Program pulse input (during OTP programming)
PWM6 / P3.4 / INT1	31	8-bit Pulse Width Modulation output6; General I/O port P3.4; External interrupt input1
VSYNCout /P3.3	32	Vertical sync output ; General output only port P3.3
HSYNCout /P3.2	33	Horizontal sync output ; General output only port P3.2
PWM5 / P2.7	34	8-bit Pulse Width Modulation output6 ; General I/O port P2.7
PWM4 / P2.6	35	8-bit Pulse Width Modulation output6 ; General I/O port P2.6
PWM3 / P2.5	36	8-bit Pulse Width Modulation output6 ; General I/O port P2.5
PWM2 / P2.4	37	8-bit Pulse Width Modulation output6 ; General I/O port P2.4
PWM1 / P2.3	38	8-bit Pulse Width Modulation output6 ; General I/O port P2.3
HSYNCin	39	Horizontal sync input
VSYNCin	40	Vertical sync input

4.2 42 SDIP



* : Open-drain option
** : Open-drain type pin

Fig. 4.1.2 42 SDIP Pinning

Table 4.2 42 SDIP Pin Descriptions

SYMBOL	PIN	DESCRIPTION
PWM0 / P2.2	1	8-bit Pulse Width Modulation output0 ; General I/O port P2.2
DPWM1 / P2.1	2	8-bit Dynamic Pulse Width Modulation output0 ; General I/O port P2.1
DPWM0 / P2.0	3	8-bit Dynamic Pulse Width Modulation output1 ; General I/O port P2.0
P3.1	4	General I/O port P3.1
P3.0	5	General I/O port P3.0
RESET	6	Reset input
VDD1	7	Power supply1 (+5V)
VSS1	8	Ground1
XTAL2	9	Oscillator output pin for system clock
XTAL1	10	Oscillator input pin for system clock
SDA2 / P1.7	11	I2C serial data I/O port ; General I/O port P1.7
SCL2 / P1.6	12	I2C serial clock I/O port ; General I/O port P1.6
P0.7	13	General I/O port P0.7 ; adapted for LED driver
P0.6	14	General I/O port P0.6 ; adapted for LED driver
P0.5	15	General I/O port P0.5
P0.4	16	General I/O port P0.4
INT0 / VPP	17	External interrupt input0 ; Programming supply voltage (during OTP programming)
P0.3	18	General I/O port P0.3
P0.2	19	General I/O port P0.2
P0.1	20	General I/O port P0.1
P0.0	21	General I/O port P0.0
ACH3 / P1.5	22	ADC channel3 input ; General I/O port P1.5
ACH2 / P1.4	23	ADC channel2 input ; General I/O port P1.4
ACH1 / P1.3	24	ADC channel1 input ; General I/O port P1.3
ACH0 / P1.2	25	ADC channel0 input ; General I/O port P1.2
SDA1 / P1.1	26	I2C serial data I/O port for the DDC interface ; General I/O port P1.1

SYMBOL	PIN	DESCRIPTION
SCL1 / P1.0	27	I2C serial clock I/O port for the DDC interface ; General I/O port P1.0
VSS2	28	Ground2
VDD2	29	Power supply2 (+5V)
SOGin / P3.7	30	Sync on Green input ; General I/O port P3.7
PATOUT / P3.6	31	Pattern out ; General I/O port P3.6
CLAMP / PWM7 / / P3.5 / PROG	32	Clamp out ; 8-bit Pulse Width Modulation output7 ; General output only port P3.5 Program pulse input (during OTP programming)
PWM6 / P3.4 / INT1	33	8-bit Pulse Width Modulation output6; General I/O port P3.4; External interrupt input1
VSYNCout /P3.3	34	Vertical sync output ; General output only port P3.3
HSYNCout /P3.2	35	Horizontal sync output ; General output only port P3.2
PWM5 / P2.7	36	8-bit Pulse Width Modulation output6 ; General I/O port P2.7
PWM4 / P2.6	37	8-bit Pulse Width Modulation output6 ; General I/O port P2.6
PWM3 / P2.5	38	8-bit Pulse Width Modulation output6 ; General I/O port P2.5
PWM2 / P2.4	39	8-bit Pulse Width Modulation output6 ; General I/O port P2.4
PWM1 / P2.3	40	8-bit Pulse Width Modulation output6 ; General I/O port P2.3
HSYNCin	41	Horizontal sync input
VSYNCin	42	Vertical sync input

5. MEMORY ORGANIZATION

The CPU manipulates operands in two memory spaces. The memory map and address spaces are shown in Fig.5.1.

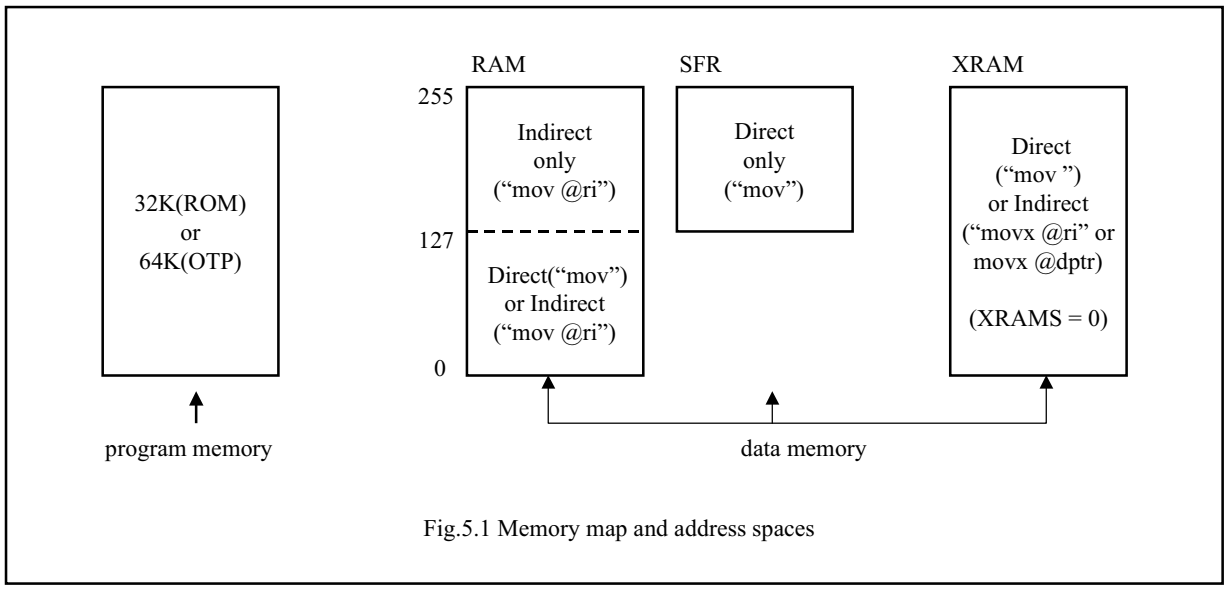


Fig.5.1 Memory map and address spaces

5.1 Program memory

The program memory consists of ROM : OTP 32K bytes (HMS97C7132) and MASK 32K bytes (HMS91C7132)

5.2 Data memory

The internal data memory is divided into four physically separated part : 256 bytes of RAM, 256 bytes of XRAM and 128 bytes of Special Function Registers (SFRs) area.

5.2.1 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 depth is only limited by the available internal RAM space of 256 bytes.

5.2.2 SFR

The SFRs can only be addressed directly in the address range from 128 to 255. Table 5.2 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 8H. The bit addresses in this area are 80H to FFH.

5.2.3 XRAM

The 256 bytes of XRAM 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 DDC Interface part). However, it is also possible to address the DRAM through MOVX command as usually used in the internal RAM extension of 80C51 derivatives.

XRAM 0 to 255 is directly addressable as external data memory locations 0 to 255 via MOVX-DPTR instruction or via MOVX-Ri instruction when the EXCON’s LSB is zero. Since external access function is not available, any access to XRAM 0 to 255 will not affect the ports.

Table 5.1 Extended control register (EXCON : 0E8H)

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

Table 5.2 SFR Memory Map

F8	-	HVGEN	CPGEN	VFH	VFL	HFH	HFL		FF
F0	*B	MDCON	MDST	VPH	HPH	VHPL			F7
E8	*EXCON					-	-	-	EF
E0	*ACC	-	-	-	-	-	-	-	E7
D8	*S1CON	S1STA	S1DAT	S1ADR0	S2CON	S2STA	S2DAT	S2ADR	DF
D0	*PSW			S1ADR1	RAMBUF	DDCDAT	DDCADR	DDCCON	D7
C8	*T2CON	-	RC2L	RC2H	-	-			CF
C0	-						-		C7
B8	*IP	-							BF
B0	*P3	DPWMCON	DPWM0	DPWM1		-	IPA	-	B7
A8	*IE	-	PWM4	PWM5	PWM6	PWM7	WDTKEY		AF
A0	*P2	PWMCON	PWM0	PWM1	PWM2	PWM3	WDTRST	IEA	A7
98	-	-							9F
90	*P1	P1SFS	P2SFS	P3SFS			ADAT	ACON	97
88	*TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	*P0	SP	DPL	DPH				PCON	87

* : bit addressable register
- : removed from the standard 8052 or reserved for the next generation

Table 5.3 Overview of the SFRs

SFR	Description	Address	Reset Value
*ACC	Accumulator	0E0H	00000000
*B	B register	0F0H	00000000
*PSW	Program Status Word	0D0H	00000000
SP	Stack Pointer	81H	00000111
DPL	Data Pointer Low	82H	00000000
DPH	Data Pointer High	83H	00000000
*P0	Port0	80H	11111111
*P1	Port1	90H	11111111
*P2	Port2	0A0H	11111111
*P3	Port3	0B0H	11111111
*IP	Interrupt Priority control	0B8H	x0000000
IPA	Interrupt Priority control	0B6H	0xxxxx00
*IE	Interrupt Enable control	0A8H	00000000
IEA	Interrupt Enable control	0A7H	0xxxxx00
TMOD	Timer/Counter Mode control	89H	00000000
*TCON	Timer/Counter Control	88H	00000000
TL0	Timer/Counter0 Low	8AH	00000000
TL1	Timer/Counter1 Low	8BH	00000000
TH0	Timer/Counter0 High	8CH	00000000
TH1	Timer/Counter1 High	8DH	00000000
*T2CON	Timer/Counter2 Control	0C8H	0xxxx0xx
RC2L	Timer/Counter2 Reload/Capture Low	0CAH	00000000
RC2H	Timer/Counter2 Reload/Capture High	0CBH	00000000
PCON	Power Control	87H	xx000000
*EXCON	Extended Control	0E8H	xxxxxxx0
P1SFS	Port1 Special Function Selection	91H	00000000
P2SFS	Port2 Special Function Selection	92H	00000000
P3SFS	Port3 Special Function Selection	93H	00000000
WDTKEY	Watchdog Timer Key	0AEH	00000000
WDTRST	Watchdog Timer Clear	0A6H	00000000

SFR	Description	Address	Reset Value
DDCCON	DDC mode status and DDC1 Control	0D7H	x00x0000
DDCADR	DDC Address pointer	0D6H	00000000
DDCDAT	DDC Data register	0D5H	00000000
RAMBUF	DDC RAM Buffer	0D4H	xxxxxxxx
*S1CON	Serial Control for DDC2	0D8H	00000000
S1STA	Serial Status for DDC2	0D9H	0x00xxxx
S1DAT	Serial Data for DDC2	0DAH	00000000
S1ADR0	Serial Address0 for DDC2	0DBH	0000000x
S1ADR1	Serial Address1 for DDC2	0D3H	0000000x
S2CON	Serial Control for I2C	0DCH	00000000
S2STA	Serial Status for I2C	0DDH	0x00xxxx
S2DAT	Serial Data for I2C	0DEH	00000000
S2ADR	Serial Address0 for I2C	0DFH	0000000x
PWMCON	PWM Control	0A1H	00000000
PWM0 - PWM3	PWM0 - PWM3 Data	0A2H-0A5H	00000000
PWM4 - PWM7	PWM4 - PWM7 Data	0AAH-0ADH	00000000
DPWMCON	Dynamic PWM Control	0B1H	0xxxxx00
DPWM0	Dynamic PWM0 Data	0B2H	00000000
DPWM1	Dynamic PWM1 Data	0B3H	00000000
MDCON	Mode Detection Control	0F1H	000xx000
MDST	Mode Detection Status	0F2H	x0000000
VPH	Vsync Period High	0F3H	00000000
HPH	Hsync Period High	0F4H	00000000
VHPL	Vsync and Hsync Period Low	0F5H	00000000
HVGEN	Hsync and Vsync Generation control	0F9H	x000x000
CPGEN	Clamp and Pattern Generation control	0FAH	00000x00
VFH	Vsync Free running High	0FBH	00100000
VFL	Vsync Free running Low	0FCH	00001010
HFH	Hsync Free running High	0FDH	01100000
HFL	Hsync Free running Low	0FEH	00x11111
ACON	ADC Control	97H	xx0x0001
ADAT	ADC Data	96H	00000000

5.3 Addressing modes

The addressing modes in the MCS-51 instruction set are as follows.

• Direct addressing
In a direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

• Indirect addressing
In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed. The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit “data pointer” register, DPTR.

• Register instructions
The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of four banks is selected at execution time by the two bank select bits in the PSW.

• Register-specific instructions
Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point it. The opcode itself does that.

• Immediate constants
The value of a constant can follow the opcode in Program memory.

• Indexed addressing
Only Program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program memory is formed by adding the Accumulator data to the base pointer.

5.4 Instruction Set

The HMS9xC7132 uses a powerful instruction set which permits the expansion of on-chip CPU peripherals and optimizes byte efficiency and execution speed. Assigned opcodes add new high-power operation and permit new addressing modes. The instruction set consists of 49 single-byte, 46 two-byte and 16 three-byte instructions. When using a 12MHz oscillator, 64 instructions execute in 1us and 45 instructions execute in 2us. Multiply and divide instructions execute in 4 us.
For the description of the Date Addressing modes and Hexadecimal opcode cross-reference see Table 5.7.

Table 5.4 Instruction set description : Arithmetic operations

Mnemonic	Description	Bytes	Cycles	Hex Code
Arithmetic operations				
ADD A, Rn	add register to A	1	1	2x
ADD A, direct	add direct byte to A	2	1	25
ADD A, @Ri	add indirect RAM to A	1	1	26, 27
ADD A, #data	add immediate data to A	2	1	24
ADDC A, Rn	add register to A with carry flag	1	1	3x
ADDC A, direct	add direct byte to A with carry flag	2	1	35
ADDC A, @Ri	add indirect RAM to A with carry flag	1	1	36, 37
ADDC A, #data	add immediate data to A with carry flag	2	1	34
SUBB A, Rn	subtract register from A with borrow	1	1	9x
SUBB A, direct	subtract direct byte from A with borrow	2	1	95
SUBB A, @Ri	subtract indirect RAM from A with borrow	1	1	96, 97
SUBB A, #data	subtract immediate data from A with borrow	1	1	94
INC A	increment A	1	1	04
INC Rn	increment register	1	1	0x
INC direct	increment direct byte	2	1	05
INC @Ri	increment indirect RAM	1	1	06, 07
DEC A	decrement A	1	1	14
DEC Rn	decrement register	1	1	1x
DEC direct	decrement direct byte	2	1	15
DEC @Ri	decrement indirect RAM	1	1	16, 17
INC DPTR	increment data pointer	1	2	A3
MUL AB	multiply A and B	1	4	A4
DIV AB	divide A by B	1	4	84
DA A	decimal adjust A	1	1	D4

Table 5.5 Instruction set description : Logical operations

Mnemonic	Description	Bytes	Cycles	Hex Code
Logical operations				
ANL A, Rn	AND register to A	1	1	5x
ANL A, direct	AND direct byte to A	2	1	55
ANL A, @Ri	AND indirect RAM to A	1	1	56, 57
ANL A, #data	AND immediate data to A	2	1	54
ANL direct, A	AND A to direct byte	2	1	52
ANL direct, #data	AND immediate data to direct byte	3	2	53
ORL A, Rn	OR register to A	1	1	4x
ORL A, direct	OR direct byte to A	2	1	45
ORL A, @Ri	OR indirect RAM to A	1	1	46, 47
ORL A, #data	OR immediate data to A	2	1	44
ORL direct, A	OR A to direct byte	2	1	42
ORL direct, #data	OR immediate data to direct byte	3	2	43
XRL A, Rn	exclusive-OR register to A	1	1	6x
XRL A, direct	exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	exclusive-OR indirect RAM to A	2	1	66, 67
XRL A, #data	exclusive-OR immediate data to A	2	1	64
XRL direct, A	exclusive-OR A to direct byte	2	1	62
XRL direct, #data	exclusive-OR immediate data to direct byte	3	2	63
CLR A	clear A	1	1	E4
CPL A	complement A	1	1	F4
RL A	rotate A left	1	1	23
RLC A	rotate A left through the carry flag	1	1	33
RR A	rotate A left	1	1	03
RRC A	rotate A right though the carry flag	1	1	13
SWAP A	swap nibbles within A	1	1	C4

Table 5.6 Instruction set description : Data transfer

Mnemonic	Description	Bytes	Cycles	Hex Code
Data transfer				
MOV A, Rn	move register to A	1	1	Ex
MOV A, direct	move direct byte to A	2	1	E5
MOV A, @Ri	move indirect RAM to A	1	1	E6, E7
MOV A, #data	move immediate data to A	2	1	74
MOV Rn, A	move A to register	1	1	Fx
MOV Rn, direct	move direct byte to register	2	2	Ax
MOV Rn, #data	move immediate data to register	2	1	7x
MOV direct, A	move A to direct byte	2	1	F5
MOV direct, Rn	move register to direct byte	2	2	8x
MOV direct, direct	move direct byte to direct	3	2	85
MOV direct, @Ri	move indirect RAM to direct byte	2	2	86, 87
MOV direct, #data	move immediate data to direct byte	3	2	75
MOV @Ri, A	move A to indirect RAM	1	1	F6, F7
MOV @Ri, direct	move direct byte to indirect RAM	2	2	A6, A7
MOV @Ri, #data	move immediate data to indirect RAM	2	1	76, 77
MOV DPTR,#data 16	load data pointer with a 16-bit constant	3	2	90
MOVC A,@A+DPTR	move code byte relative to DPTR to A	1	2	93
MOVC A, @A+PC	move code byte relative to PC to A	1	2	83
MOVX A, @Ri	move external RAM (8-bit address) to A	1	2	E2, E3
MOVX A, @DPTR	move external RAM (16-bit address) to A	1	2	E0
MOVX @Ri, A	move A to external RAM (8-bit address)	1	2	F2, F3
MOVX @DPTR, A	move A to external RAM (16-bit address)	1	2	F0
PUSH direct	push direct byte onto stack	2	2	C0
POP direct	pop direct byte from stack	2	2	D0
XCH A, Rn	exchange register with A	1	1	Cx
XCH A, direct	exchange direct byte with A	2	4	C5
XCH A, @Ri	exchange indirect RAM with A	1	1	C6, C7
XCHD A, @Ri	exchange LOW-order digit indirect RAM with A	1	1	D6, D7

Table 5.7 Instruction set description: Boolean variable manipulation, Program branching

Mnemonic	Description	Bytes	Cycles	Hex Code
Boolean variable manipulation				
CLR C	clear carry flag	1	1	C3
CLR bit	clear direct bit	2	1	C2
SETB C	set carry flag	1	1	D3
SETB bit	set direct bit	2	1	D2
CPL C	complement carry flag	1	1	B3
CPL bit	complement direct bit	2	1	B2
ANL C, bit	AND direct bit to carry flag	2	1	82
ANL C, /bit	AND complement of direct bit to carry flag	2	2	B0
ORL C, bit	OR direct bit to carry flag	2	2	72
ORL C, /bit	OR complement of direct bit to carry flag	2	2	A0
MOV C, bit	move direct bit to carry flag	2	1	A2
MOV bit, C	move carry flag to direct bit	2	2	92
JC rel	jump if carry flag is set	2	2	40
JNC rel	jump if carry flag is not set	2	2	50
JB bit, rel	jump if direct bit is set	3	2	20
JNB bit, rel	jump if direct bit is not set	3	2	30
JBC bit, rel	jump if direct bit is set and clear bit	3	2	10
Program branching				
ACALL addr11	absolute subroutine call	2	2	y1
LCALL addr16	long subroutine call	3	1	12
RET	return from subroutine	1	2	22
RETI	return from interrupt	2	2	32
AJMP addr16	absolute jump	2	2	z1
LJMP addr16	long jump	3	2	02
SJMP rel	short jump (relative address)	2	2	80
JMP @A+DPTR	jump indirect relative to the DPTR	1	2	73
JZ rel	jump if A is zero	2	2	60
JNZ rel	jump if A is not zero	2	2	70
CJNE A, direct, rel	compare direct to A and jump if not equal	3	2	B5
CJNE A, #data, rel	compare direct to A and jump if not equal	3	2	B4
CJNE Rn, #data, rel	compare immediate to register and jump if not equal	3	2	Bx
CJNE @Ri,#data,rel	compare immediate indirect and jump if not equal	3	2	B6, B7
DJNZ Rn, rel	decrement register and jump if not zero	2	2	Dx
DJNZ direct, rel	decrement direct and jump if not zero	3	2	D5
NOP	no operation	1	1	00

Table 5.8 Description of the mnemonics in the instruction set

Mnemonic	Description
Data addressing modes	
Rn	working register R0-R7
direct	128 internal RAM locations and any special function register (SFR)
@Ri	indirect internal RAM location addressed by register by register R0 or R1 of the actual register bank
#data	8-bit constant included in instruction
#data16	16-bit constant included as bytes 2 and 3 of instruction
bit	direct addressed bit in internal RAM or SFR
addr16	16-bit destination address. Used by LCALL and LJMP; the branch will be anywhere within the 64 kbytes Program Memory address space
addr11	111-bit destination address. Used by ACALL and AJMP;the branch will be within the same 2 kbytes page of Program Memory as the first byte of the following instruction
rel	signed(two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps; range is - 128 to + 127 bytes relative to first byte of the following instruction
Hexadecimal opcode cross-reference	
x	8, 9, A, B, C, D, E, F
y	1, 3, 5, 7, 9, B, D, F
z	0, 2, 4, 6, 8, A, C, E

6. INTERRUPT SYSTEM

There are interrupt requests from 9 sources as follows.

- INT0 external interrupt
- Timer0 interrupt
- Timer1 interrupt
- Timer2 interrupt
- INT1 external interrupt
- DDC interrupt
- MD interrupt
- VSYNC interrupt
- I2C interrupt

6.1 Interrupt sources

6.1.1 INT0 external interrupt

- The INT0 can be either level-active or transition-active depending on bit IT0 in register TCON. The flag that actually generates this interrupt is bit IE0 in TCON
- When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.
- If the interrupt was level-activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

6.1.2 INT1 external interrupt

- The INT1 can be either level-active or transition-active depending on bit IT1 in register TCON. The flag that actually generates this interrupt is bit IE1 in TCON
- When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.
- If the interrupt was level-activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

6.1.3 MD interrupt

- A MD interrupt is generated by the hardware mode detector in case of mode change, horizontal or vertical.
- This flag has to be cleared by the software.

6.1.4 VSYNC interrupt

- The changing of the VSYNC level can generate an interrupt. This depends on the setting that is programmed in the MDINT-SFR. Via this register it is possible to enable the edge of the VSYNC-signal that should generate the interrupt. Both edges can be controlled separately.
- The interrupt flag has to be cleared by the software.

6.1.5 DDC interrupt

- The DDC interrupt is generated either by bit INTR in the S1STA register for DDC2B/DDC2AB/DDC2B+ protocol or by bit DDC_int in the DDCCON register for DDC1 protocol or by bit SWHINT bit in the DDCCON register when DDC protocol is changed from DDC1 to DDC2
- Flags except the INTR have to be cleared by the software. INTR flag is cleared by hardware.

6.1.6 I2C interrupt

- The interrupt of the second I2C is generated by bit INTR in the register S2STA.
- This flag is cleared by hardware.

6.1.7 Timer0 and Timer1 interrupt

- Timer0 and Timer1 interrupts are generated by TF0 and TF1 which are set by an overflow of their respective Timer/Counter registers(except for Timer0 in mode3).
- These flags are cleared by the internal hardware.

6.1.8 Timer2 interrupt
• Timer2 interrupt is generated by TF2 which is set by an overflow of Timer2.
• This flag has to be cleared by the software.

All of the 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.

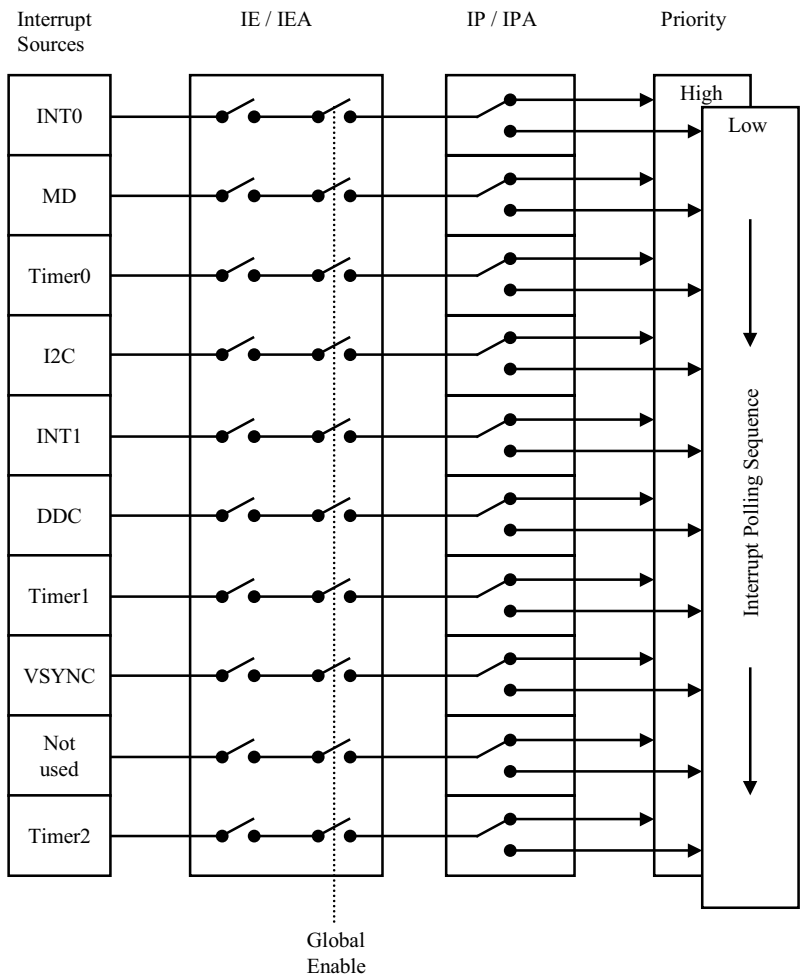


Fig. 6.1 Interrupt System

6.2 Interrupt Enable structure

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function register IE and IEA. All interrupt source can also be globally disabled by clearing bit EA in IE.

Table 6.1 Interrupt Enable Register (IE : 0A8H)

7	6	5	4	3	2	1	0
EA	EVSYNC	ET2	ES	ET1	EX1	ET0	EX0

Table 6.2 Description of the IE bits

BIT	SYMBOL	FUNCTION
7	EA	Disable all interrupts. 0 : no interrupt will be acknowledged 1 : each interrupt source is individually enabled or disabled by setting or clearing its enable bit
6	EVSYNC	Enable Vsync interrupt
5	ET2	Enable timer2 interrupt
4	ES	Not used
3	ET1	Enable timer1 interrupt
2	EX1	Enable external interrupt (INT1)
1	ET0	Enable timer0 interrupt
0	EX0	Enable external interrupt (INT0)

Table 6.3 Interrupt Enable Register (IEA : 0A7H)

7	6	5	4	3	2	1	0
EDDC	-	-	-	-	-	EI2C	EMD

Table 6.4 Description of the IEA bits

BIT	SYMBOL	FUNCTION
7	EDDC	Enable DDC interrupt
6	EX6	Not used
5	EX5	Not used
4	EX4	Not used
3	EX3	Not used
2	EX2	Not used
1	EI2C	Enable I2C interrupt
0	EMD	Enable MD interrupt

6.3 Interrupt Priority structure

Each interrupt source can be assigned one of two priority levels.
Interrupt priority levels are defined by the interrupt priority special function register IP and IPA.
"0" = low priority
"1" = high priority
A low priority interrupt may be interrupted by a high priority interrupt level interrupt. A high priority interrupt routine cannot be interrupted by any other interrupt source. If two interrupts of different priority occur simultaneously, the high priority level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 6.7.

Table 6.5 Priority levels

SOURCE	PRIORITY WITHIN LEVEL
INT0	1 (highest)
MD	
Timer0	
I2C	
INT1	
DDC	
Timer1	
VSYNC	9 (lowest)
Timer2	

Note
- The "Priority within level" structure is only used to resolve simultaneous requests of the same priority level.
- The MD interrupt needs a higher priority then ALL the other interrupts. This is to avoid that a mode change will not be serviced in time and that the setting of the S-curve is not updated in time. When the S-curve settings are not updated in time (after a mode change) the monitor may be damaged.

Table 6.6 Interrupt Priority Register (IP : 0B8H)

7	6	5	4	3	2	1	0
-	PVSYNC	PT2	PS	PT1	PX1	PT0	PX0

Table 6.7 Description of the IE bits

BIT	SYMBOL	FUNCTION
7	-	Reserved
6	PVSYNC	Vsync interrupt priority level
5	PT2	Timer2 interrupt priority level
4	PS	Not used
3	PT1	Timer1 interrupt priority level
2	PX1	External interrupt (INT1) priority level
1	PT0	Timer0 interrupt priority level
0	PX0	External interrupt (INT0) priority level

Table 6.8 Interrupt Priority Register (IPA : 0B6H)

7	6	5	4	3	2	1	0
PDDC	-	-	-	-	-	PI2C	PMD

Table 6.9 Description of the IPA bits

BIT	SYMBOL	FUNCTION
7	PDDC	DDC interrupt priority level
6	PX6	Not used
5	PX5	Not used
4	PX4	Not used
3	PX3	Not used
2	PX2	Not used
1	PI2C	I2C interrupt priority level
0	PMD	MD interrupt priority level

6.4 How Interrupt are handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during 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 H/W generated LCALL is not blocked by any of the following conditions :

- An interrupt of equal priority or higher priority level is already in progress.
- The current machine cycle is not the final cycle in the execution of the instruction in progress.
- The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers.

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 being responded to for one of the above mentioned conditions, if the flag is still inactive when the blocking condition is removed, 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 processor acknowledges an interrupt request by executing a hardware generated LCALL to the appropriate service 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 6.10.

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 6.10 Vector addresses

SOURCE	VECTOR ADDRESS
INT0	0003H
MD	004BH
Timer0	000BH
I2C	0043H
INT1	0013H
DDC	003BH
Timer1	001BH
VSYNC	0033H
Timer2	002BH

7. POWER-SAVING MODE

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

The following functions are switched off when the microcontroller enters the Idle mode.
• CPU (halted)
• I2C interface (halted)
• PWM0 to PWM7 and DPWM0 to DPWM2 (reset, output = High)
• 8-bit ADC (aborted if conversion in progress)

The following functions remain active during Idle mode. These functions may generate an interrupt or reset and thus terminate the Idle mode.
• Timer0, Timer1 and Timer2
• Watchdog timer
• DDC interface
• External interrupt
• Mode detection

In Power-down mode, the system clock is halted. Both the oscillator will be stopped after setting the bit PD in PCON.

7.1 Power control register

The modes Idle and Power-down are activated by software via the PCON register.

Table 7.1 Power control register (PCON : 87H)

7	6	5	4	3	2	1	0
-	-	LVREN	LVRLS	GF1	GF0	PD	IDL

Table 7.2 Description of the PCON bits

BIT	SYMBOL	FUNCTION
7 to 6	-	Not used
5	LVREN	Enable low voltage reset
4	LVRLS	Select low VDD level ; 3.7V or 3.5V
3	GF1	General purpose flag bit
2	GF0	General purpose flag bit
1	PD	Activate Power-down mode
0	IDL	Activate Idle mode

Table 7.3 External Pin Status During Idle and Power-down modes

MODE	MEMORY	PORT0 - 3	SYNC	PWM	I2C	DDC	-	-
Idle	Internal	Data	on	High	High-Z	on	-	-
Power-down	Internal	Data	High	High	High-Z	High-Z	-	-

7.2 Idle mode

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 : Stack pointer, Program counter, Program status word, Accumulator, RAM and All other registers maintain their data during idle mode.
There are three ways to terminate the idle mode.

- Activation of any enabled interrupt X0, T0, X1, T1 etc. 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.
- External hardware reset : the hardware reset is required to be active for two machine cycle to complete the reset operation.
- Internal watchdog reset : the microcontroller restarts after 3 machine cycles in all cases.

7.3 Power-down mode

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 contents of the on-chip RAM and the Special Function Register are preserved.

The power-down mode can be terminated by an external RESET in the same way as in the 80C51 (but SFRs are cleared due to RESET).

8. INPUT/OUTPUT (I/O)

The HMS9xC7132 has four 8-bit ports (Port0, Port1, Port2 and Port3).
Port0 - Port3 are the same as in the 80C51, with the exception of the additional functions of Port1, Port2 and Port3.
All ports are bidirectional and Pins of which the alternative function is not used may be used as normal bidirectional I/Os except Port3.2, Port3.3 and Port3.5(These Pins can be only used as the output).
The use of Port1- Port3 pins as alternative functions are carried out automatically by the HMS9xC7132 provided the associated SFR bit is set HIGH.
Port0 is the type of open-drain I/O. Port0.6 and Port0.7 have the capability to drive LED.
Fig. 8.1 shows the port structure.

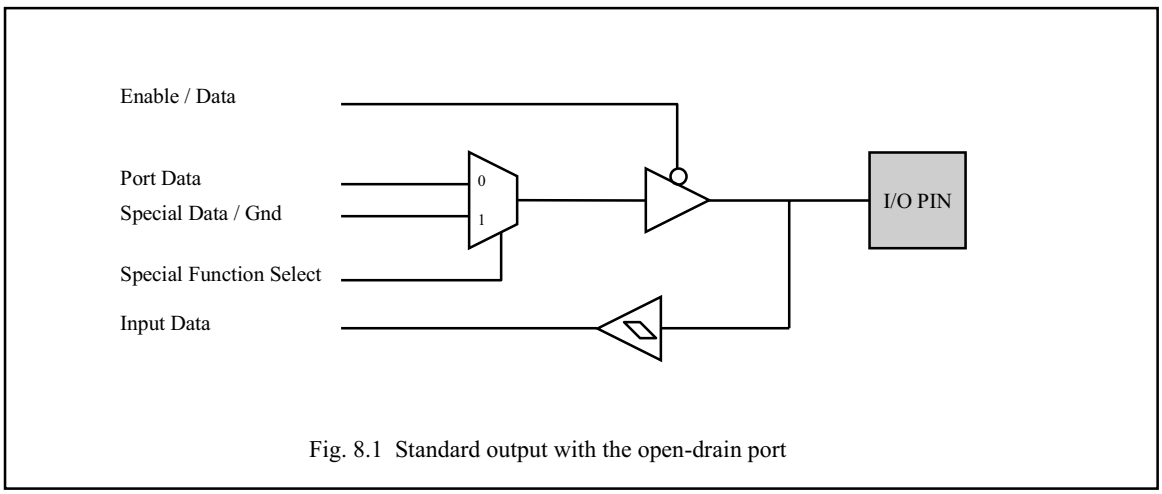


Fig. 8.1 Standard output with the open-drain port

The alternative function for Port1, Port2 and Port3 can be described as follows.

- Port 0 : No alternative function.
- Port 1 : P1.0 is combined with the SCL1 interface line(open-drain)
  P1.1 is combined with the SDA1 interface line (open-drain)
  P1.2 is combined with the ACH0 interface line (high-z)
  P1.3 is combined with the ACH1 interface line (high-z)
  P1.4 is combined with the ACH2 interface line (high-z)
  P1.5 is combined with the ACH3 interface line (high-z)
  P1.6 is combined with the SCL2 interface line (open-drain)
  P1.7 is combined with the SDA2 interface line (open-drain)
- Port 2 : P2.0 is combined with the dynamic PWM0 interface line(open-drain or push-pull)
  P2.1 is combined with the dynamic PWM1 interface line (open-drain or push-pull)
  P2.2 is combined with the static PWM0 interface line (open-drain or push-pull)
  P2.3 is combined with the static PWM1 interface line (open-drain or push-pull)
  P2.4 is combined with the static PWM2 interface line (open-drain or push-pull)
  P2.5 is combined with the static PWM3 interface line (open-drain or push-pull)
  P2.6 is combined with the static PWM4 interface line (open-drain or push-pull)
  P2.7 is combined with the static PWM5 interface line (open-drain or push-pull)
- Port 3 : P3.0 has not alternative function.
  P3.1 has not alternative function.
  P3.2 is combined with the HSYNCout interface line (push-pull)
  P3.3 is combined with the VSYNCout interface line (push-pull)
  P3.4 is combined with the PWM6 interface line (open-drain or push-pull)
  P3.5 is combined with the CLAMP or PWM7 interface line (push-pull)
  P3.6 is combined with the PATOUT interface line (push-pull)
  P3.7 is combined with the SOG interface line (pull-up)

8.1 Pin function selection

Several SFRs are used to select the port-function or the alternative function of the external pin.

Table 8.3 Port1 special function selection register (P1SFS : 91H)

7	6	5	4	3	2	1	0
P1SFS7	P1SFS6	P1SFS5	P1SFS4	P1SFS3	P1SFS2	P1SFS1	P1SFS0

Table 8.4 Description of the P1SFS bits

BIT	SYMBOL	FUNCTION	RESET
7	P1SFS7	The selection of the pin function. 0 : pin 11 has P1.7 function. 1 : pin 11 has SDA2 function.	0
6	P1SFS6	The selection of the pin function. 0 : pin 12 has P1.6 function. 1 : pin 12has SCL2 out function.	0
5	P1SFS5	The selection of the pin function. 0 : pin 22 has P1.5 function. 1 : pin 22 has ACH3 function.	0
4	P1SFS4	The selection of the pin function. 0 : pin 23 has P1.4 function. 1 : pin 23 has ACH2 function.	0
3	P1SFS3	The selection of the pin function. 0 : pin 24 has P1.3 function. 1 : pin 24 has ACH1 function.	0
2	P1SFS2	The selection of the pin function. 0 : pin 25 has P1.2 function. 1 : pin 25 has ACH0 function.	0
1	P1SFS1	The selection of the pin function. 0 : pin 26 has P1.1 function. 1 : pin 26 has SDA1 function.	0
0	P1SFS0	The selection of the pin function. 0 : pin 27 has P1.0 function. 1 : pin 27 has SCL1 function.	0

Table 8.5 Port2 special function selection register (P2SFS : 92H)

7	6	5	4	3	2	1	0
P2SFS7	P2SFS6	P2SFS5	P2SFS4	P2SFS3	P2SFS2	P2SFS1	P2SFS0

Table 8.6 Description of the P2SFS bits

BIT	SYMBOL	FUNCTION	RESET
7	P2SFS7	The selection of the pin function. 0 : pin 36 has P2.7 function 1 : pin 36 has PWM5 out function.	0
6	P2SFS6	The selection of the pin function. 0 : pin 37 has P2.6 function 1 : pin 37 has PWM4 out function.	0
5	P2SFS5	The selection of the pin function. 0 : pin 38 has P2.5 function. 1 : pin 38 has PWM3 out function.	0
4	P2SFS4	The selection of the pin function. 0 : pin 39 has P2.4 function. 1 : pin 39 has PWM2 out function.	0
3	P2SFS3	The selection of the pin function. 0 : pin 40 has P2.3 function. 1 : pin 40 has PWM1 out function.	0
2	PP2SFS2	The selection of the pin function. 0 : pin 1 has P2.2 function. 1 : pin 1 has PWM0 out function.	0
1	P2SFS1	The selection of the pin function. 0 : pin 2 has P2.1 function. 1 : pin 2 has DPWM1 out function.	0
0	P2SFS0	The selection of the pin function. 0 : pin 3 has P2.0 function. 1 : pin 3 has DPWM0 out function.	0

Table 8.7 Port3 special function selection register (P3SFS : 93H)

7	6	5	4	3	2	1	0
P3SFS7	P3SFS6	P3SFS5	P3SFS4	P3SFS3	P3SFS2	P3SFS1	P3SFS0

Table 8.8 Description of the P3SFS bits

BIT	SYMBOL	FUNCTION	RESET
7	P3SFS7	The selection of the pin function. 0 : pin 30 has P3.7 function 1 : pin 30 has SOG input function.	0
6	P3SFS6	The selection of the pin function. 0 : pin 31 has P3.6 function 1 : pin 31 has PATOUT function.	0
5	P3SFS5	The selection of the pin function. 0 : pin 32 has P3.5 function. 1 : pin 32 has CLAMP or PWM7 function.	0
4	P3SFS4	The selection of the pin function. 0 : pin 33 has P3.4 function. 1 : pin 33 has PWM6 function.	0
3	P3SFS3	The selection of the pin function. 0 : pin 34 has P3.3 function. 1 : pin 34 has VSYNCout function.	0
2	P3SFS2	The selection of the pin function. 0 : pin 35 has P3.2 function. 1 : pin 35has HSYNCout function.	0
1	P3SFS1	The selection of the pin function. 0 : pin 4 has P3.1 function. 1 : reserved	0
0	P3SFS0	The selection of the pin function. 0 : pin 5has P3.0 function. 1 : reserved	0

9. OSCILLATOR

The oscillator circuit of the HMS9xC7132 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. To drive the HMS9xC7132 externally, XTAL1 is driven from an external source and XTAL2 left open-circuit.

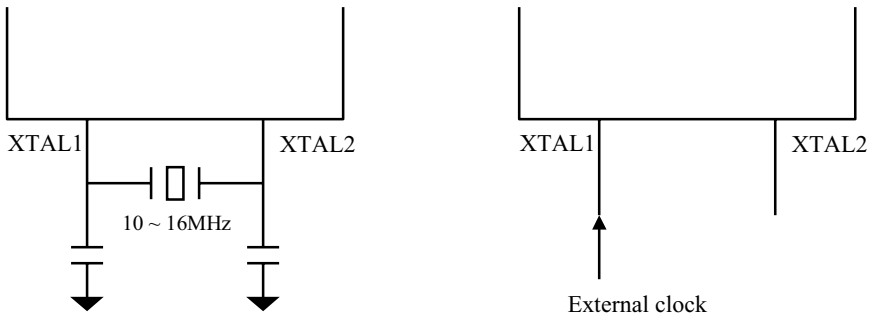


Fig. 9.1 Oscillator configurations

10. RESET

There are three ways to invoke a reset and initialize the HMS9xC7132.
• Via the external RESET pin
• Via the Watchdog Timer overflow
• Via low VDD voltage reset

The reset mechanism is illustrated in Figure 10.1.

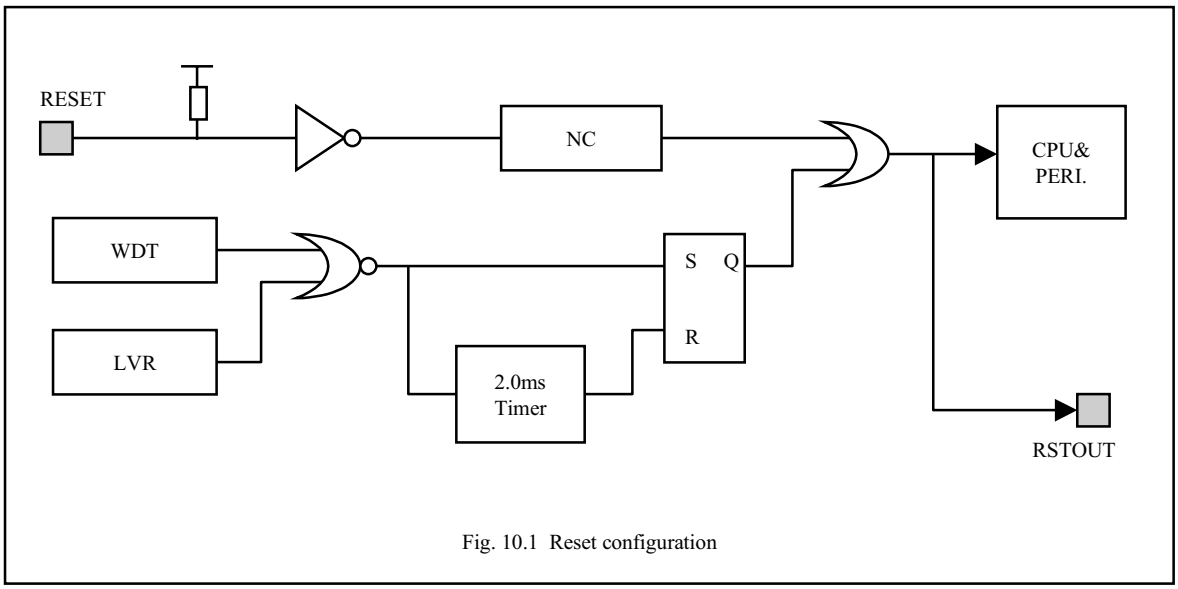


Fig. 10.1 Reset configuration

Each reset source will cause an internal reset signal active. The CPU responds by executing an internal reset and puts the internal registers in a defined state.

10.1 External reset

The reset pin RESET is connected to a Schmitt trigger for noise reduction. A reset is accomplished by holding the RESET pin LOW for at least 2 machine cycles (24 system clock), while the oscillator is running.

An automatic reset can be obtained by switching on VDD, if the RESET pin is connected to GND via a capacitor and to the VDD via resistor. The capacitor should be at least 10uF.
The increase of the RESET pin voltage depends on the capacitor. The voltage must remain below the higher threshold for at minimum the oscillator start-up time plus 2 machine cycles.

10.2 Watchdog timer overflow

The length of the output pulse from the WDT is over 2048 machine cycles. In chapter 11, the watchdog timer is described in more detail.

10.3 Low VDD voltage reset

When VDD is below 3.7V, the built-in low voltage detector generates an internal reset signals. The reset signal will be LOW during 2ms @12MHz after the voltage is higher than 3.7V.