HE8P1700A SERIES TARGET SPECIFICATION

1. GENERAL DESCRIPTION 4

2. FEATURES 4

3. PAD ASSIGNMENT 5

4. BLOCK DIAGRAM 8

4.1 PIN DESCRIPTION 9

5. PROGRAM MEMORY (ROM) 9

6. DATA MEMORY (RAM) 10

6.1 RAM BANK LOCATION 10

6.2 RAM BANK SELECTION 10

6.3 SYSTEM REGISTER ARRANGEMENT (BANK 0) 10

7. ACCUMULATOR 13

7.1 CARRY FLAG 13

7.2 DECIMAL CARRY FLAG 13

7.3 ZERO FLAG 13

8. WORKING REGISTERS 13

8.1 H, L REGISTERS 13

8.2 Y, Z REGISTERS 14

8.3 LOOK-UP TABLE 14

8.4 ADDRESSING MODE 14

9. PROGRAM COUNTER 15

9.1 ONE ADDRESS SKIPPING 15

9.2 MULTI-ADDRESS JUMPING 15

10. STACK BUFFER 16

10.1 ACC & WORKING REGISTERS PROTECTION 16

11. OSCILLATOR 17

11.1 OSCM REGISTER 17

11.2 INTERNAL LOW CLOCK 18

11.3 HIGH-LOW CLOCK EXCHANGE 18

11.4 0.5 SECOND RESTART FUNCTION 19

12. GTMR PRESCALER 19

12.1 WARMUP TIME 19

12.2 WATCH DOG (WDOG) TIMER 20

13. BASIC TIMER (T0) 21

13.1 T0M MODE REGISTER 21

13.2 T0C COUNTING REGISTER 21

14. TIMER/EVENT COUNTER (TC0) 22

14.1 TC0M MODE REGISTER 22

14.2 TC0C COUNTING REGISTER 23

14.3 TC0R AUTO-LOAD REGISTER 23

14.4 PWM0 FUNCTION DESCRIPTION 23

15. TIMER/EVENT COUNTER (TC1) 24

15.1 TC1M MODE REGISTER 24

15.2 TC1C COUNTING REGISTER 25

15.3 TC1R AUTO-LOAD REGISTER 25

15.4 PWM1 FUNCTION DESCRIPTION 25

16. SERIAL INPUT/OUTPUT TRANSCEIVER (SIO) 26

16.1 SIOM MODE REGISTER 26

16.2 SIOB/SIOR DATA BUFFER 26

17. INTERRUPT 27

17.1 INTEN INTERRUPT ENABLE REGISTER 27

17.2 INTRQ INTERRUPT REQUEST REGISTER 27

18. I/O PORT 28

18.1 PORT 1 WAKEUP (P1W) REGISTER 28

18.2 PULL-UP RESISTOR (PNUR) REGISTER 29

18.3 PORT MODE (PNM) REGISTER 29

18.4 PORT (PN) DATA REGISTER 29

19. 8-CHANNEL ANALOG TO DIGITAL CONVERTER 29

19.1 ADM REGISTER 29

19.2 ADB & ADR REGISTERS 29

19.3 ADC TEST MODE 30

20. 7-BIT DIGITAL TO ANALOG CONVERTER 30

21. ABSOLUTE MAXIMUM RATING 32

22. ELECTRICAL CHARACTERISTIC 32

22.1 HE8P1700A SERIES OTP OSCILLATOR FREQUENCY vs. OPERATING VOLTAGE GRAPH 33

23.INSTRUCTION SET 34

24.PACKAGE INFORMATION 35

HE8P1700A 8-bit microcontroller with 8-bit ADC circuit

1. GENERAL DESCRIPTION

The HE8P1700A is an 8-bit micro-controller utilized with CMOS technology fabrication and featured with low power consumption and high performance by its unique electronic structure. This chip is designed with the excellent IC structure, including the program memory up to 4K-word OTP ROM, 256 bytes of the data memory, one 8-bit T0 basic timer, two 8-bit timer/event counters, a watchdog timer, seven interrupt sources (T0, TC0, TC1, SIO, INT0 ~ INT2), an 8-ch ADC converter with 8 bits resolution, a 7-bit DAC converter, 2-ch PWM output, 33 I/O pins and 8 levels stack buffer. Besides, the user can choose desired oscillator configurations for the controller. There are four oscillator configurations to select for generating system clock, including Hi/Low Speed crystal, ceramic resonator or cost-saving RC.

2. FEATURES

◆ Memory configuration
OTP ROM size: 4096 * 16 bits.
RAM size: 256 * 8 bits.

◆ I/Opin configuration (Total 33 pins)
One input port: 3 pins with wakeup function.
Three Input/output ports: 22 pins for general purpose.
One input/output port: 8 pins shared with ADC inputs.

◆ An 8-bit basic timer.

◆ Two 8-bit timer/event counters.

◆ 59 powerful instructions
All of instructions are 1 word with 1 or 2 cycles' execution.
Execution time: One cycle uses 4 clocks of oscillator.
All ROM area JMP instruction.
All ROM area Subroutine CALL instruction.
All ROM area lookup table function. (MOVC instruction)

MUL instruction for arithmetic multiplication.

◆ Seven interrupt sources:
Four internal interrupts: TO, TC0, TC1, SIO
Three external interrupts: 1NT0 - INT2

◆ Eight levels stack buffer

◆ An 8-channel ADC converter with 12-bit resolution.

◆A DAC converter with 7 bits resolution.

◆ Two 8-bit PWM output (PWM0,PWM1)

◆ Two Buzzer output (TC0,TC1)

◆ A watchdog timer.

◆ Built-in internal RC Low Clock 16KHz
Crystal type: speed up to 20 MHz.
RC type: speed up to 10MHz.

◆ Package:
PDIP48, SSOP48
QFP44
PDIP40
SKDIP28
SOP28
SSOP20
PDIP 18
SOP18

SN8P1702:

◆ Memory configuration
OTP ROM size: 1024 * 16 bits.
RAM size: 64 * 8 bits.
I/O pin configuration (Total 12 pins)
One input port: 1 pins with wakeup function.
Two Input/output ports: 7 pins for general purpose.
One input/output port: 4 pins shared with ADC inputs.

◆ An 8-bit timer/event counters.

◆ 59powerful instructions
All of instructions are 1 word with 1 or 2 cycles' execution.
Execution time: One cycle uses 4 clocks of oscillator.
All ROM area JMP instruction.
All ROM area Subroutine CALL instruction.
All ROM area lookup table function. (MOVC instruction)

◆ Two interrupt sources:
One internal interrupts: TC0
One external interrupts: INT0

◆ Eight levels stack buffer

◆ An 4-channel ADC converter with 12-bit resolution.

◆ One 8-bit PWM output (PWM0)

◆ One Buzzer output (TC0)

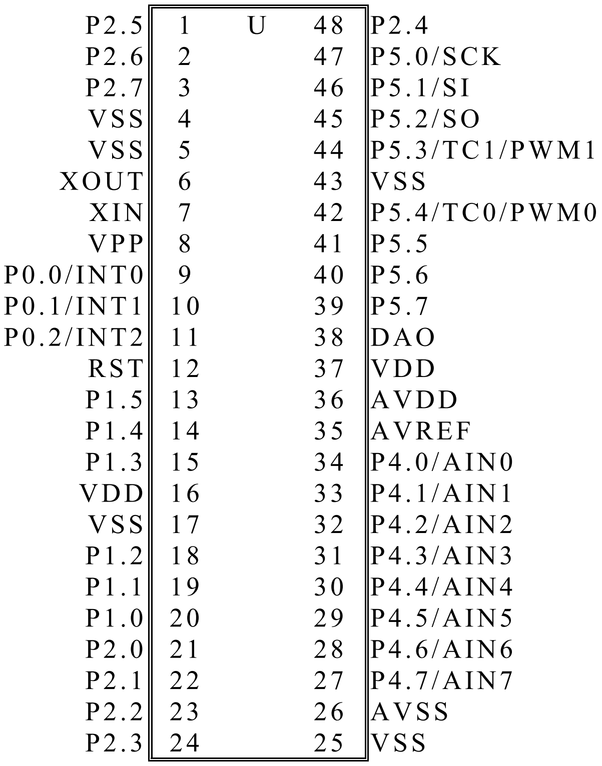
◆ A watchdog timer.

◆ Built-in internal RC Low Clock 16KHz
Crystal type: speed up to 20 MHz.
RC type:speed up to 10MHz.

◆ Package:
SSOP20
PDIP 18
SOP18

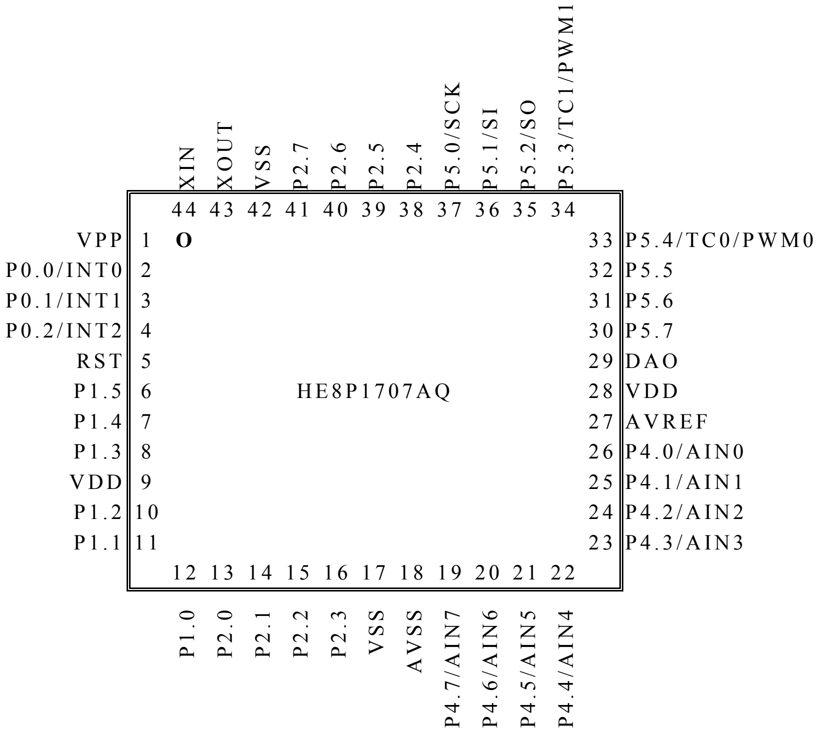
3. PAD ASSIGNMENT

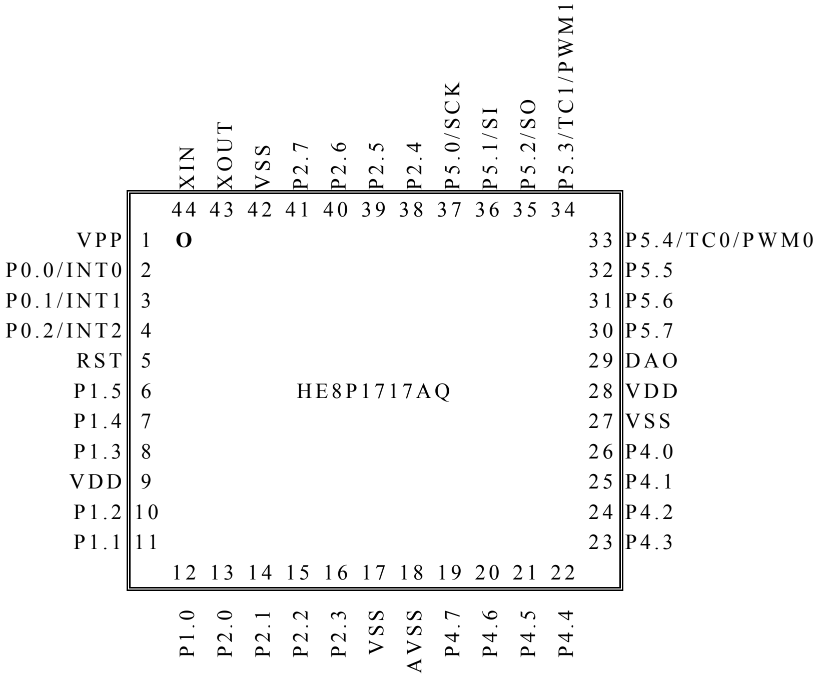
48PIN：



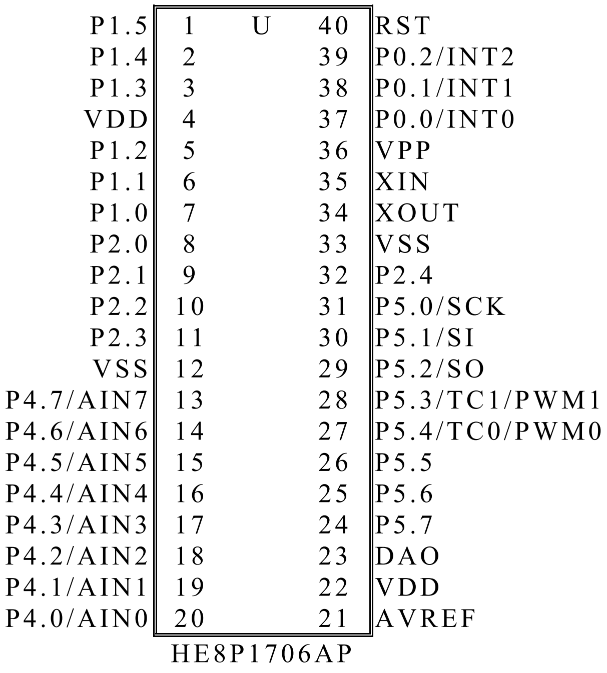
HE8P1708AP
HE8P1708AX X:SSOP

44PIN：



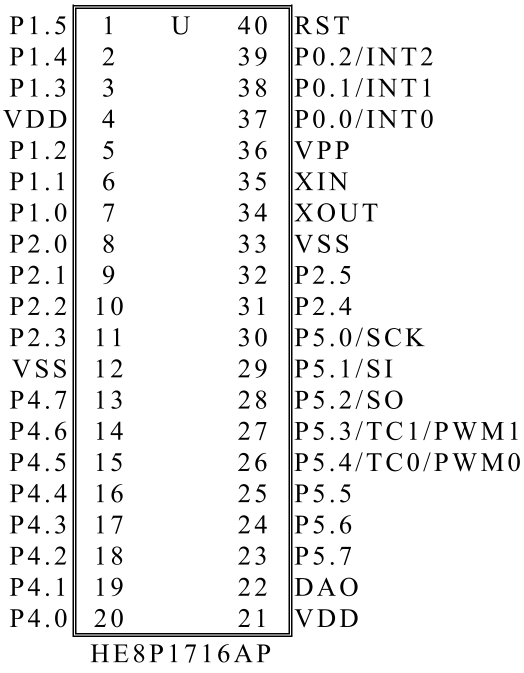


40PIN：



P1.5	1	40	RST
P1.4	2	39	P0.2/INT2
P1.3	3	38	P0.1/INT1
VDD	4	37	P0.0/INT0
P1.2	5	36	VPP
P1.1	6	35	XIN
P1.0	7	34	XOUT
P2.0	8	33	VSS
P2.1	9	32	P2.4
P2.2	10	31	P5.0/SCK
P2.3	11	30	P5.1/SI
VSS	12	29	P5.2/SO
P4.7/AIN7	13	28	P5.3/TC1/PWM1
P4.6/AIN6	14	27	P5.4/TC0/PWM0
P4.5/AIN5	15	26	P5.5
P4.4/AIN4	16	25	P5.6
P4.3/AIN3	17	24	P5.7
P4.2/AIN2	18	23	DAO
P4.1/AIN1	19	22	VDD
P4.0/AIN0	20	21	AVREF

HE8P1706AP



P1.5	1	40	RST
P1.4	2	39	P0.2/INT2
P1.3	3	38	P0.1/INT1
VDD	4	37	P0.0/INT0
P1.2	5	36	VPP
P1.1	6	35	XIN
P1.0	7	34	XOUT
P2.0	8	33	VSS
P2.1	9	32	P2.5
P2.2	10	31	P2.4
P2.3	11	30	P5.0/SCK
VSS	12	29	P5.1/SI
P4.7	13	28	P5.2/SO
P4.6	14	27	P5.3/TC1/PWM1
P4.5	15	26	P5.4/TC0/PWM0
P4.4	16	25	P5.5
P4.3	17	24	P5.6
P4.2	18	23	P5.7
P4.1	19	22	DAO
P4.0	20	21	VDD

HE8P1716AP

28PIN：

		U		
P1.4	1		28	RST
P1.3	2		27	P0.2/INT2
VDD	3		26	P0.1/INT1
P1.2	4		25	P0.0/INT0
P1.1	5		24	VPP
P1.0	6		23	XIN
VSS	7		22	XOUT
P4.4/AIN4	8		21	VSS
P4.3/AIN3	9		20	P5.0/SCK
P4.2/AIN2	10		19	P5.1/SI
P4.1/AIN1	11		18	P5.2/SO
P4.0/AIN0	12		17	P5.3/TC1/PWM1
AVREF	13		16	P5.4/TC0/PWM0
VDD	14		15	DAO

HE8P1704AK
HE8P1704AS

		U		
P1.5	1		28	RST
P1.4	2		27	P0.2/INT2
P1.3	3		26	P0.1/INT1
VDD	4		25	P0.0/INT0
P1.2	5		24	VPP
P1.1	6		23	XIN
P1.0	7		22	XOUT
VSS	8		21	VSS
P4.4	9		20	P5.0/SCK
P4.3	10		19	P5.1/SI
P4.2	11		18	P5.2/SO
P4.1	12		17	P5.3/TC1/PWM1
P4.0	13		16	P5.4/TC0/PWM0
VDD	14		15	DAO

HE8P1714AK
HE8P1714AS

20PIN：

		U		
P5.1/SI	1		20	P5.2/SO
P5.0/SCK	2		19	P5.3/TC1/PWM1
XOUT	3		18	P5.4/TC0/PWM0
XIN	4		17	VDD
VPP	5		16	VDD
P0.0/INT0	6		15	P4.0/AIN0
RST	7		14	P4.1/AIN1
P1.1	8		13	P4.2/AIN2
P1.0	9		12	P4.3/AIN3
VSS	10		11	VSS

HE8P1702AX

X:SSOP

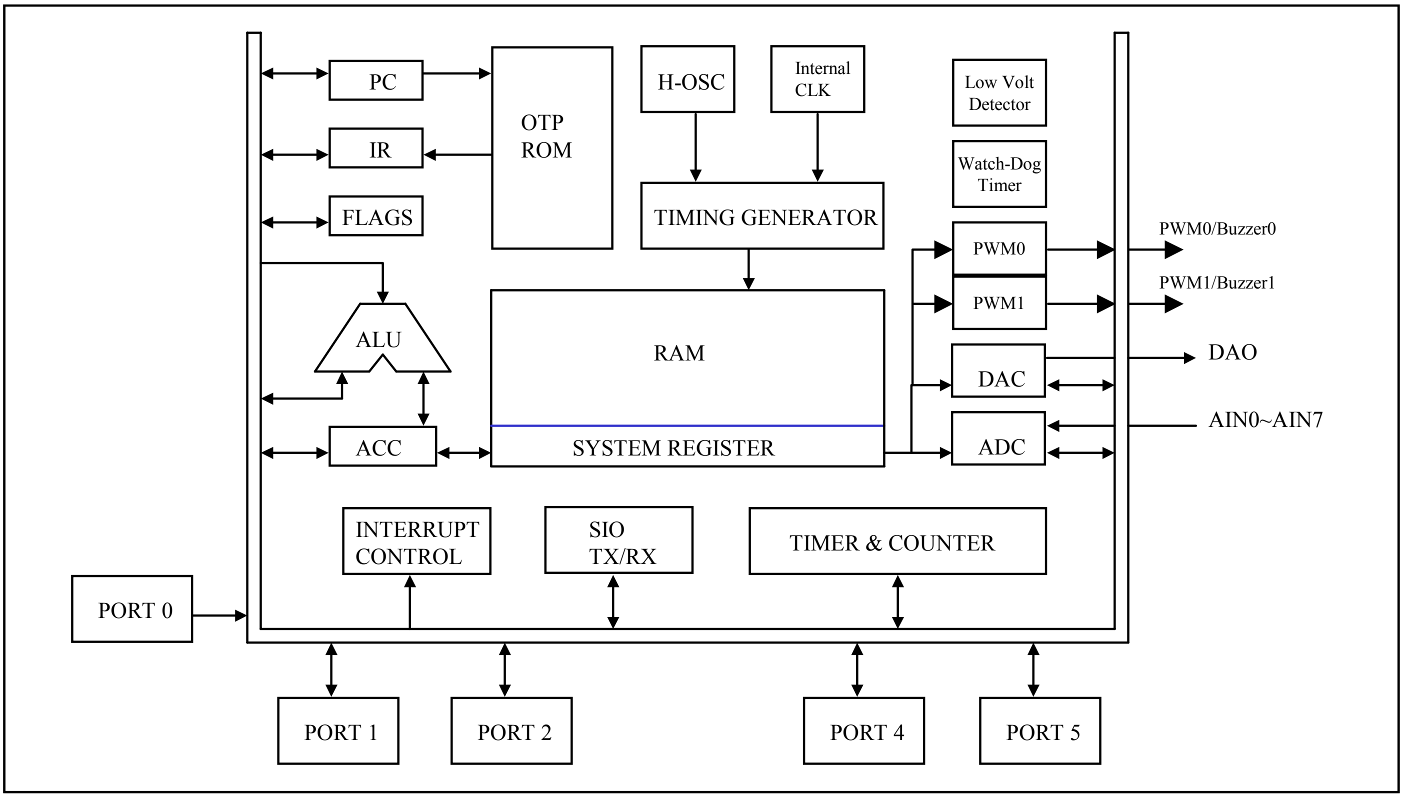
18PIN：

		U		
P0.0/INT0	1		18	VPP
RST	2		17	XIN
P1.1	3		16	XOUT
P1.0	4		15	P5.0/SCK
VSS	5		14	P5.1/SI
P4.3/AIN3	6		13	P5.2/SO
P4.2/AIN2	7		12	P5.3/TC1/PWM1
P4.1/AIN1	8		11	P5.4/TC0/PWM0
P4.0/AIN0	9		10	VDD

HE8P1702AP
HE8P1702AS

4. BLOCK DIAGRAM

S86PA system block



4.1 PIN DESCRIPTION

PIN NAME	TYPE	DESCRIPTION
VDD, VSS	P	Power supply input pins.
VPP/NC	P	During program op-code, this pin be pull to 12.5Vdc to reset internal address counter and to write data into OTP-ROM. This pin must be kept no connection during normal operation.
RST	I	System reset inputs pin. Schmitt trigger structure, active “low”, normal stay to “high”.
D0 ~ D7	I	OTP programming data input pins shared with Port 4 and Port 5.
XIN, XOUT	I, O	Oscillator pins.
P0.0 / INT0	I	Port 0.0 and INT0 trigger pin with wakeup function. (Schmitt trigger structure)
P0.1 / INT1	I	Port 0.1 and INT1 trigger pin with wakeup function. (Schmitt trigger structure)
P0.2 / INT2	I	Port 0.2 and INT2 trigger pin with wakeup function. (Schmitt trigger structure)
P1.0 ~ P1.5	I/O	Port 1.0~ Port 1.5 bi-direction pins , all port 1 with wakeup function.
P2.0 ~ P2.7	I/O	Port 2.0 ~ Port 2.7 bi-direction pins.
P4.0 ~ P4.7	I/O	Port 4.0 ~ Port 4.7 bi-direction pins.
P5.0 /SCK	I/O	Port 5.0 bi-direction pin and SIO’s clock input/output.
P5.1/SI	I/O	Port 5.1 bi-direction pin and SIO’s data input.
P5.2/SO	I/O	Port 5.2 bi-direction pin and SIO’s data output.
P5.3/TC1/PWM1	I/O	Port 5.3 bi-direction pin , TC1 ÷ 2 signal output pin or PWM1 output pin.
P5.4/TC0/PWM0	I/O	Port 5.4 bi-direction pin , TC0 ÷ 2 signal output pin or PWM0 output pin.
P5.5 ~ P5.7	I/O	Port 5.5 ~ Port 5.7 bi-direction pins.
AVREF	I	A/D converter analog reference voltage.
AIN0 ~ AIN7	I	Analog signal input pins for ADC converter.
DAO	O	DAC signal output pin.

5.PROGRAM MEMORY (ROM)

The HE8P1700A provides the program memory up to 4K-word to be addressed and is able to fetch instructions through 12-bit wide PC (Program Counter). It also can lookup ROM data by using ROM code registers (R, X, Y, Z). All of the program memory is partitioned into two coding areas, located from 0000H to 000FH and from 0010H to 0FFFH. The former area is assigned for executing interrupt vector. And the later area is for storing instruction’s OP-code and lookup table’s data. *The last location (0FFFH) of OTP ROM had been reserved, it can not be used by programming.*

	OTP ROM
0000h	Reset vector
0001h	
0002h	
0003h	
0004h	Reserved
0005h	“
0006h	“
0007h	“
0008h	Interrupt vector
0009h	.
000Ah	.
.	.
0010h	General purpose area
.	.
.	.
.	.
0FFEh	Bottom of OTP user Rom.
0FFFh	Reserved

4K * 16 OTP Rom Size

6. DATA MEMORY (RAM)

The HE8P1700A has built-in 256-bytes (00H ~ 7FH, 100~17FH) data memory to store general purpose data. Beside this, memory location from 80H to FFH are assigned for special registers application.

6.1 RAM BANK LOCATION

	RAM location		
000H	General purpose area	RAM	bank 0
-	-	RAM	bank 0
07FH	bottom of bank 0	RAM	bank 0
080h	system register	Register	
	-	Register	
0FFH	end of system register	Register	
100H	Band 1	RAM	bank 1
	-	RAM	bank 1
	-	RAM	bank 1
17FH	end of Bank 1	RAM	bank 1

6.2 RAM BANK SELECTION

The RBANK is a 1-bit register located at 87H in RAM bank_0. The user can select RAM bank by using this register pointing to working RAM bank for ACC to read/write RAM data.

RBANK initial value = xxxx xxx0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RBANK	-	-	-	-	-	-	-	RBNKS0

RBNKS0 : RAM bank selects bits. 0 = RAM Bank 0 , 1 = RAM Bank 1

6.3 SYSTEM REGISTER ARRANGEMENT (BANK 0)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8	L	H	R	Z	Y	X	PFLAG	RBANK	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B	DAM	ADM	ADB	ADR	SIOM	SIOR	SIOB	-	-	-	-	-	-	-	-	-
C	P1W	P1M	P2M	-	P4M	P5M	-	-	INTRQ	INTEN	OSCM	-	-	TC0R	PCL	PCH
D	P0	P1	P2	-	P4	P5	-	-	T0M	T0C	TC0M	TC0C	TC1M	TC1C	TC1R	STKP
E	P0UR	P1UR	P2UR	-	P4UR	P5UR	@HL	@YZ	-	-	-	-	-	-	-	-
F	STK7	STK7	STK6	STK6	STK5	STK5	STK4	STK4	STK3	STK3	STK2	STK2	STK1	STK1	STK0	STK0

L, H =	Working & @HL addressing register	R =	Working register and ROM lookup data buffer
X =	Working and ROM address register	Y, Z =	Working, @YZ and ROM addressing register
PFLAG =	ROM page and special flag register	RBANK=	RAM Bank Select register
ADM =	ADC’s mode register	ADR =	ADC’s resolution selects register
DAM =	DAC’s mode register	ADB =	ADC’s data buffer
P1W =	Port 1 wakeup register	PnM =	Port n input/output mode register
PnUR =	Port n pull-up register	Pn =	Port n data buffer
INTRQ =	Interrupts’ request register	INTEN =	Interrupts’ enable register
OSCM =	Oscillator mode register	PCH, PCL =	Program counter
T0M =	Timer 0 mode register	TC0M =	Timer/Counter 0 mode register
T0C =	Timer 0 counting register	TC0C =	Timer/Counter 0 counting register
TC1M =	Timer/Counter 1 mode register	TC0R =	Timer/Counter 0 auto-reload data buffer
TC1C =	Timer/Counter 1 counting register	TC1R =	Timer/Counter 1 auto-reload data buffer
		STKP =	Stack pointer buffer
SIOM =	SIO mode control register	SIOB =	SIO’s data buffer
SIOR =	SIO’s clock reload buffer	STK0~STK7 =	Stack 0 ~ stack 7 buffer
@HL =	RAM HL indirect addressing index pointer	@YZ =	RAM YZ indirect addressing index pointer

System register table

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Remarks
080H	LBIT7	LBIT6	LBIT5	LBIT4	LBIT3	LBIT2	LBIT1	LBIT0	R/W	L
081H	HBIT7	HBIT6	HBIT5	HBIT4	HBIT3	HBIT2	HBIT1	HBIT0	R/W	H
082H	RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0	R/W	R
083H	ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0	R/W	Z
084H	YBIT7	YBIT6	YBIT5	YBIT4	YBIT3	YBIT2	YBIT1	YBIT0	R/W	Y
085H	XBIT7	XBIT6	XBIT5	XBIT4	XBIT3	XBIT2	XBIT1	XBIT0	R/W	X
086H	-	-	-	-	-	C	DC	Z	R/W	PFLAG
087H	-	-	-	-	-	-	-	RBNKS0	R/W	RBANK
0B0H	DAENB	DAB6	DAB5	DAB4	DAB3	DAB2	DAB1	DAB0	R/W	DAM data register
0B1H	ADENB	ADS	EOC	GCHS	-	CHS2	CHS1	CHS0	R/W	ADM mode register
0B2H	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	R	ADB data buffer
0B3H	-	ADCKS	ADLEN	TADC	ADB3	ADB2	ADB1	ADB0	R/W	ADR register
0B4H	SENB	START	SRATE1	SRATE0	-	SCKMD	SEDGE	TXRX	R/W	SIOM mode register
0B5H	SIOR7	SIOR6	SIOR5	SIOR4	SIOR3	SIOR2	SIOR1	SIOR0	W	SIOR reload buffer
0B6H	SIOB7	SIOB6	SIOB5	SIOB4	SIOB3	SIOB2	SIOB1	SIOB0	R/W	SIOB data buffer
0B7H	-	-	-	-	-	-	-	-	-	-

(To be continued)

System register table

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Remarks
0C0H	-	-	P15W	P14W	P13W	P12W	P11W	P10W	W	P1W wakeup register
0C1H	-	-	P15M	P14M	P13M	P12M	P11M	P10M	R/W	P1M I/O direction
0C2H	P27M	P26M	P25M	P24M	P23M	P22M	P21M	P20M	R/W	P2M I/O direction
0C3H	-	-	-	-	-	-	-	-	-	-
0C4H	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M	R/W	P4M I/O direction
0C5H	P57M	P56M	P55M	P54M	P53M	P52M	P51M	P50M	R/W	P5M I/O direction
0C6H	-	-	-	-	-	-	-	-	-	-
0C7H	-	-	-	-	-	-	-	-	-	-
0C8H	-	TC1IRQ	TC0IRQ	T0IRQ	SIOIRQ	P02IRQ	P01IRQ	P00IRQ	R/W	INTRQ
0C9H	-	TC1IEN	TC0IEN	T0IEN	SIOIEN	P02IEN	P01IEN	P00IEN	R/W	INTEN
0CAH	WTCKS	WDRST	WDRate	-	CPUM0	CLKMD	STPHX	-	R/W	OSCM
0CBH	-	-	-	-	-	-	-	-	-	-
0CCH	-	-	-	-	-	-	-	-	-	-
0CDH	TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0	W	TC0R
0CEH	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	R/W	PCL
0CFH	-	-	-	-	PC11	PC10	PC9	PC8	-	PCH
0D0H	-	-	-	-	-	P02	P01	P00	R	P0 data buffer
0D1H	-	-	P15	P14	P13	P12	P11	P10	R/W	P1 data buffer
0D2H	P27	P26	P25	P24	P23	P22	P21	P20	R/W	P2 data buffer
0D3H	-	-	-	-	-	-	-	-	-	-
0D4H	P47	P46	P45	P44	P43	P42	P41	P40	R/W	P4 data buffer
0D5H	P57	P56	P55	P54	P53	P52	P51	P50	R/W	P5 data buffer
0D6H	-	-	-	-	-	-	-	-	-	-
0D7H	-	-	-	-	-	-	-	-	-	-
0D8H	T0ENB	T0rate2	T0rate1	T0rate0	-	-	-	-	R/W	T0M
0D9H	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0	R/W	T0C
0DAH	TC0ENB	TC0rate2	TC0rate1	TC0rate0	TC0CKS	ALOAD0	TC0OUT	PWM0OUT	R/W	TC0M
0DBH	TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0	R/W	TC0C
0DCH	TC1ENB	TC1rate2	TC1rate1	TC1rate0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT	R/W	TC1M
0DDH	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0	R/W	TC1C
0DEH	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0	W	TC1R
0DFH	GIE	-	-	-	STKPB3	STKPB2	STKPB1	STKPB0	R/W	STKP stack pointer
0E0H	-	-	-	-	-	P02R	P01R	P00R	W	P0UR
0E1H	-	-	P15R	P14R	P13R	P12R	P11R	P10R	W	P1UR
0E2H	P27R	P26R	P25R	P24R	P23R	P22R	P21R	P20R	W	P2UR
0E3H	-	-	-	-	-	-	-	-	-	-
0E4H	P47R	P46R	P45R	P44R	P43R	P42R	P41R	P40R	W	P4UR
0E5H	P57R	P56R	P55R	P54R	P53R	P52R	P51R	P50R	W	P5UR
0E6H	@HL7	@HL6	@HL5	@HL4	@HL3	@HL2	@HL1	@HL0	R/W	@HL index pointer
0E7H	@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0	R/W	@YZ index pointer
0F0H	S7PC7	S7PC6	S7PC5	S7PC4	S7PC3	S7PC2	S7PC1	S7PC0	R	STK7L
0F1H	-	-	-	-	S7PC11	S7PC10	S7PC9	S7PC8	R	STK7H
0F2H	S6PC7	S6PC6	S6PC5	S6PC4	S6PC3	S6PC2	S6PC1	S6PC0	R	STK6L
0F3H	-	-	-	-	S6PC11	S6PC10	S6PC9	S6PC8	R	STK6H
“	“	“	“	“	“	“	“	“	“	“
“	“	“	“	“	“	“	“	“	“	“
“	“	“	“	“	“	“	“	“	“	“
0FCH	S1PC7	S1PC6	S1PC5	S1PC4	S1PC3	S1PC2	S1PC1	S1PC0	R	STK1L
0FDH	-	-	-	-	S1PC11	S1PC10	S1PC9	S1PC8	R	STK1H
0FEH	S0PC7	S0PC6	S0PC5	S0PC4	S0PC3	S0PC2	S0PC1	S0PC0	R	STK0L
0FFH	-	-	-	-	S0PC11	S0PC10	S0PC9	S0PC8	R	STK0H

Note : In the above table, it will get a logic “H” data, when use instruction to read or test null position.

7. ACCUMULATOR

The ACC is an 8-bits data register responsible for transferring or manipulating data between ALU and data memory. If the result of operating appears to be zero (Z) or occurs carry (C or DC), then these flags will be set to PFLAG register.

PFLAG initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	-	-	-	-	-	C	DC	Z

7.1 CARRY FLAG

C = 1 : If executed arithmetic addition with occurring carry signal or executed arithmetic subtraction without borrowing signal or executed rotation instruction with shifting out logic "1".
C = 0 : If executed arithmetic addition without occurring carry signal or executed arithmetic subtraction with borrowing signal or executed rotation instruction with shifting out logic "0".

7.2 DECIMAL CARRY FLAG

DC = 1 : If executed arithmetic addition with occurring signal from low nibble or executed arithmetic subtraction without borrow signal from high nibble.
DC = 0 : If executed arithmetic addition without occurring signal from low nibble or executed arithmetic subtraction with borrow signal from high nibble.

7.3 ZERO FLAG

Z = 1 : After operation, the content of ACC is zero.
Z = 0 : After operation, the content of ACC is not zero.

8. WORKING REGISTERS

The locations 80H to 86H of RAM bank 0 in data memory stores the specially defined registers such as register H, L, R, X, Y, Z and PFLAG, respectively shown in the following table. These registers can use as the general purpose of working buffer and can also be used to access ROM's and RAM's data. For instance, all of the ROM's table can be looked-up with R, X, Y and Z registers. And the data of RAM memory can be indirectly accessed with H, L , Y and Z registers.

RAM	80H	81H	82H	83H	84H	85H	86H	87H
	L	H	R	Z	Y	X	PFLAG	-

8.1 H, L REGISTERS

The H and L are 8-bit register with two major functions. One is to use the registers as working register. The other is to use the registers as data pointer to access RAM's data. The @HL that is data point_0 index buffer located at address E6H in RAM bank_0. It employs H and L registers to addressing RAM location in order to read/write data through ACC. The Lower 4-bit of H register is pointed to RAM bank number and L register is pointed to RAM address number, respectively. The higher 4-bit data of H register is truncated in RAM indirectly access mode.

Example: If want to read a data from RAM address 20H of bank_0, it can use indirectly addressing mode to access data as following

```
B0MOV   H,#00H          ; To set RAM bank 0 for H register
B0MOV   L,#20H          ; To set location 20H for L register
B0MOV   A,@HL           ; To read a data into ACC
```

8.2 Y, Z REGISTERS

The Y and Z registers are the 8-bit buffers. There are three major functions of these registers. First, Y and Z registers can be used as working registers. Second, these two registers can be used as data pointers for @YZ register. Third, the registers can be address ROM location in order to lookup ROM data.

8.3 LOOK-UP TABLE

In the ROM's data lookup function, the X register is pointed to the highest 8-bit, Y register to the middle 8-bit and Z register to the lowest 8-bit data of ROM address. After MOVC instruction is executed, the low-byte data of ROM then will be stored in ACC and high-byte data stored in R register.

Example: To lookup ROM's data from location table_1

```
                B0MOV   X,#TABLE1$H             ; To set lookup table1's high address
                B0MOV   Y, #TABLE1$M            ; To set lookup table's middle address.
                B0MOV   Z, #TABLE1$L            ; To set lookup table's low address.
                MOVC                            ; To lookup data,   R = 00H,   ACC = 35H
                .                               ;
                INCMS   Z                       ; To lookup next ROM's data
                NOP                             ;
                MOVC                            ; To lookup data,   R = 51H,   ACC = 05H
                .       .                       ;
TABLE1:         DW      0035H                   ; To define a word (16 bits) data
                DW      5105H                   ;   "
                DW      2012H                   ;   "
```

8.4 ADDRESSING MODE

The HE8P1700A provides three addressing modes to access RAM data, including immediate mode, direct mode and indirect mode. The main purpose of these three different modes are described in the following table. The immediate addressing mode uses an immediate data to set up the location in (MOV A,#I,   B0MOV M,#I) in ACC or specific RAM location. The directly addressing mode uses address number to access memory location (MOV A,12H,   MOV 12H,A). The indirectly addressing mode is set an address in data pointer registers (H/L or Y/Z) and uses MOV instruction to read/write data between ACC and @HL/@YZ register (MOV A,@HL,   MOV @HL,A).

Immediate addressing mode

```
                MOV     A,#12H             ; To set an immediate data 12H into ACC
                B0MOV   X,#28H             ; To set an immediate data 28H into X register
```

Directly addressing mode

```
                B0MOV   A,12H              ; To get a content of location 12H of bank 0 and save in ACC
```

Indirectly addressing mode with @HL/@YZ register

```
                CLR     H                  ; To clear H register
                B0MOV   L,#12H             ; To set an immediate data 12H into L register
                B0MOV   A,@HL              ; Use data pointer @HL reads a data from RAM location 012H into ACC
;
                CLR     Y                  ; To clear Y register
                B0MOV   Z,#16H             ; To set an immediate data 16H into Z register
                B0MOV   A,@YZ              ; Use data pointer @YZ reads a data from RAM location 016H into ACC
```

9. PROGRAM COUNTER

The program counter (PC) is a 12-bit binary counter separated into the high-byte 4 bits and the low-byte 8 bits. This counter is responsible for pointing a location in order to fetch an instruction for kernel circuit. Normally, the program counter is automatically incremented with each instruction during program execution. Besides, it can be replaced with specific address by executing CALL or JMP instruction. When JMP or CALL instruction is executed, the destination address will be inserted to bit 0 ~ bit 11.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCH	-	-	-	-	PC11	PC10	PC9	PC8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

PC Initial value = xxxx 0000 0000 0000

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0

9.1 ONE ADDRESS SKIPPING

If the condition of bit test instruction is match, the PC will add 2 steps to skip next instruction.

```
            B0BTS1   FC                 ; To skip, if Carry_flag = 1
            JMP      C0STEP             ;   else jump to C0STEP.
            .
            .
C0STEP:     NOP
```

9.2 MULTI-ADDRESS JUMPING

Users can jump round multi-address by either JMP instruction or ADD M,A instruction (M = PCL) to activate multi-address jumping function. If carry signal occurs after execution of ADD PCL,A, the carry signal will not affect PCH register.

```
Example : If PC = 0323H   (PCH = 03H 、PCL = 23H)
;PC = 0323H
            MOV      A,#28H
            B0MOV    PCL,A              ; Jump to address 0328H
            .        .
;PC = 0328H .        .
            MOV      A,#00H
            B0MOV    PCL,A              ; Jump to address 0300H
```

```
Example : If PC = 0323H   (PCH = 03H 、PCL = 23H)
;PC = 0323H
            B0ADD    PCL,A              ; PCL = PCL + ACC, the PCH can not be changed.
            JMP      A0POINT            ; If ACC = 0, jump to A0POINT
            JMP      A1POINT            ;   ACC = 1, jump to A1POINT
            JMP      A2POINT            ;   ACC = 2, jump to A2POINT
            JMP      A3POINT            ;   ACC = 3, jump to A3POINT
            .        .                  ;
```

10. STACK BUFFER

The stack buffer has up to 8-level areas and each level is 12-bit length. This buffer is designed to store PC's value while the interrupt service or call subroutine is executed. The STKP register is a pointer designed to point active level in order for kernel circuit to push or pop up PC's data from stack buffer. The STKP will decrease one level after the data is pushed into stack buffer and increase one level before data is popped up from stack buffer. Once interrupt occurs, the global interrupts will turn the enable bit (GIE) of STKP to be disable. And GIE will turn to be enable again, after RETI instruction is executed.

STKP initial value = 0xxx 1111

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKP	GIE	-	-	-	STKPB3	STKPB2	STKPB1	STKPB0

STKn initial value = xxxx xxxx xxxx xxxx,   STKn = STKnH + STKnL (n = 7H ~ 0H)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnH	-	-	-	-	SnPC11	SnPC10	SnPC9	SnPC8

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnL	SnPC7	SnPC6	SnPC5	SnPC4	SnPC3	SnPC2	SnPC1	SnPC0

10.1 ACC & WORKING REGISTERS PROTECTION

The HE8P1700A does not push ACC and working registers into stack buffer during interrupt execution. Thus, once interrupt occurs, these data must be stored in the data memory based on the user's program as follows:

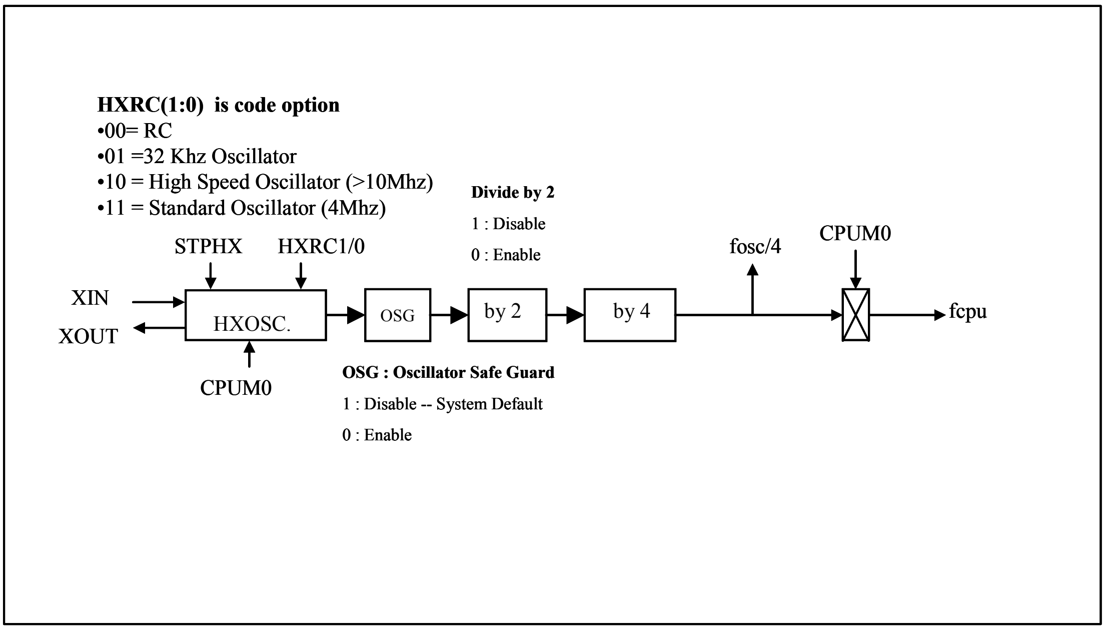
```
; To declare variables in source program
ACCBUF          EQU        00H                       ; To declare ACC_buffer at 00H in bank 0

Example : To push ACC and working registers
PUSHBUF:        PUSH                                 ; To push working registers and flag register
                B0MOV      ACCBUF,A                  ; To store ACC after enter the interrupt

Example : To pop ACC and working registers
POPBUF:         B0XCH      A,ACCBUF                  ; To re-load ACC before exit the interrupt
                POP                                  ; To pop working registers and flag register
```

11. OSCILLATOR

The HE8P1700A uses single oscillator, which can be RC, crystal or ceramic resonator, to generate system clock source. The user can select desired one of them to be the oscillator configuration of the chip. The chip featured with low power consumption by using its power down mode to turn the system into idle situation. The HE8P1700A will switch the system from normal mode to power down mode by setting STPHX = 1. On the contrast, the chip can be awakened from the power down mode into normal mode by triggering Port 0 or Port 1. After the system wakeups, the STPHX bit then will reset to 0 automatically.



11.1 OSCM REGISTER

OSCM initial value = 000x 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	WTCKS	WDRST	WDrate	-	CPUM0	CLKMD	STPHX	HXWUP

WTCKS : Watch-dog clock source select bit. 0 = fcpu , 1 : internal RC low Clock
CLKMD : High / Low speed mode select bit. 0 = normal (dual) mode, 1 = internal RC mode
HXWUP: High speed oscillator's warm up time control bit. 0 = 18th, 1 = 15th.
STPHX : To stop high-speed oscillator control bit. 0 = free run, 1 = stop.
CPUM0: CPU operating mode control bit. 0 = operating, 1 = power down mode, turn off both the external and the internal clock

Note : To recommend execution a NOP instruction after changing CPU operating mode..

Example : To switch cpu operating from normal to power down mode.

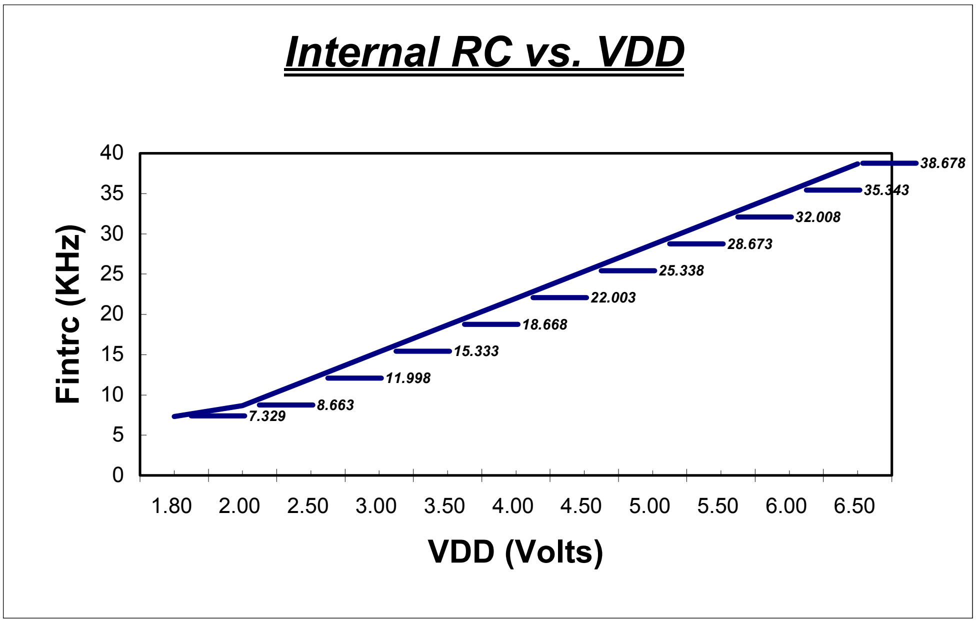
```
N2SLEEP:      B0BCLR   FHXWUP          ; To setup warm up time for next system wakeup.
              B0BSET   FCPUM0          ; To switch cpu from normal to power down mode.
              NOP
              NOP
```

Operating mode description

MODE	NORMAL	Internal Low Clock	SLEEP	REMARK
HX osc.	Running	By STPHX	Stop	
CPU instruction	Executing	Executing	Stop	
T0 timer	Active	Active	Inactive	* Active can be pro-
TC0/TC1 timer	Active	Active	Inactive	Programming.
WDOG timer	Active	Active	Inactive	Programming.
Internal interrupt	All active	All Active	All inactive	
External interrupt	All active	All active	All inactive	
Wakeup source	-	-	P0, P1, Reset	P0 & P1 accept "L"

11.2INTERNAL LOW CLOCK

HE8P1700A builds in a internal RC clock. It is a low clock for slow mode. Under slow mode, all instructions still run as normal mode. The frequency is about 16 KHz, and the power consumption is just only 15uA in VDD = 3V status (external oscillator stops). CLKMD bit of OSCM register is to switch system high/low clock. CLKMD = 0 is high clock (external oscillator). CLKMD = 1 is low clock (internal RC). The characteristic between RC and VDD is as following.



11.3HIGH-LOW CLOCK EXCHANGE

System is in slow mode (internal low clock). The external oscillator stops (STPHX = 1). If system goes to normal mode, users have to make a delay time for external oscillator stable by programming. The delay time is equal to 1/Fosc * 2048. External oscillator frequency is 3.58MHz. The routine is as following.

```
; Normal mode to slow mode routine
        B0BSET      FCLKMD          ; Switch system clock to low clock (internal RC)
        B0BSET      FSTPHX          ; Stop high clock (external oscillator)
        .           .

; Slow mode to normal mode routine

        B0BCLR      FSTPHX          ; Enable external oscillator (high clock)

        NOP                         ; Wakeup time routine
        NOP
        NOP

        B0BCLR      FCLKMD          ; Swtich to normal mode (high speed mode)
        .           .
        .           .
        .           .
```

1/Fosc * 2048 of @3.58MHz is equal to 0.57ms. One instruction cycle of internal low clock (16KHz) is equal to 0.25ms. 1/Fosc * 2048 is about 3 instruction cycles of internal low clock. That is 3 **NOP**s.

11.4 0.5 SECOND RESTART FUNCTION

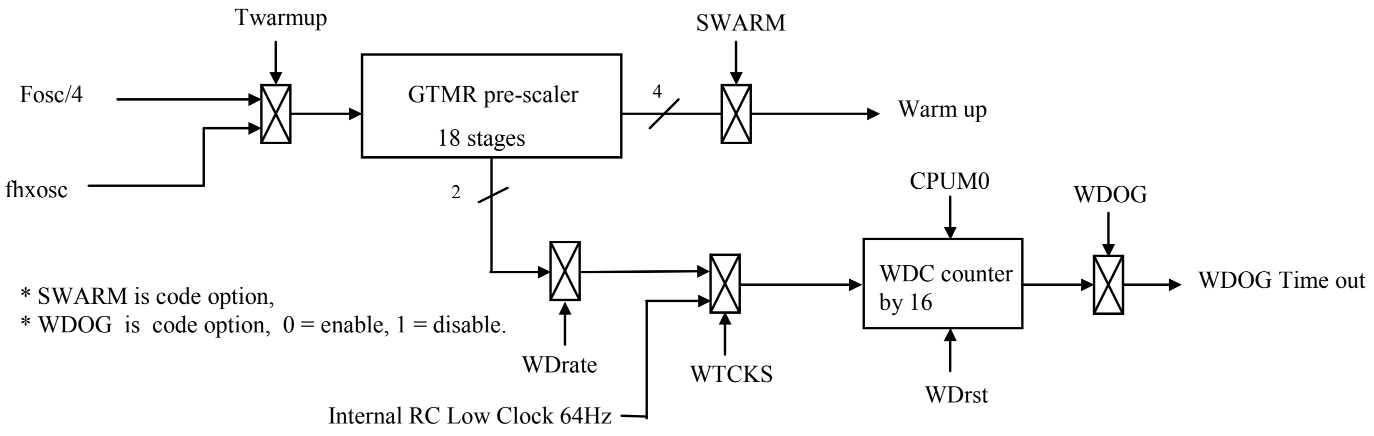
Combining internal RC and watch dog can make 0.5sec restart function. This function is applied to power saving status and other applications using 0.5sec timer. Select watch dog clock source to internal low clock. The watch dog clock is 32Hz. And make a routine let watch dog timer overflow. The program will restart from ORG0. The overflow time is 0.5sec in VDD = 3V. The demo program is as following.

```
; 0.5sec restart routine

        B0BSET      FWTCKS              ; Set watch dog clock is internal RC
        B0BSET      FWDRST              ; Reset watch dog timer
        .           .
        .           .
        .           .                   ; Watch dog timer overflow and system restart from ORG0
```

12. GTMR PRESCALER

The GTMR pre-scaler is an 18-bit binary up counter designed to count a precision timing for watchdog timer and HX oscillator warm up time applications. There are two clock sources for GTMR pre-scaler selection. One is Fosc/4 for all of function operation and the other is fhxosc for operating CPU warm up.



* SWARM is code option,
* WDOG is code option, 0 = enable, 1 = disable.

12.1 WARMUP TIME

There are six warm-up times designed for application with different type of oscillator. One is fhxosc $\div$ 2^{18} and fhxosc $\div$ 2^{15} designed for high-speed oscillator application. And the other is fhxosc $\div$ 2^{14} and fhxosc $\div$ 2^{11} designed for low-speed oscillator application. In actual application, user can set SWARM control bit and HXWUP bit of OSCM register to select suitable warm up time to meet oscillator's start-up time before stopping high speed oscillator.

The following warm up time is mapping to HX_osc.

HX_osc	SWARM	HXWUP	Warm up time
3.58 MHz	0	0	1 / (fhxosc $\div$ 2^{18}) = 73.2 ms
		1	1 / (fhxosc $\div$ 2^{15}) = 9 ms
400 KHz	1	0	1 / (fhxosc $\div$ 2^{14}) = 40.9 ms
		1	1 / (fhxosc $\div$ 2^{11}) = 5 ms
40 KHz		0	1 / (fhxosc $\div$ 2^{14}) = 40.9 ms
		1	1 / (fhxosc $\div$ 2^{11}) = 50 ms

12.2 WATCH DOG (WDOG) TIMER

The watchdog timer (WDTMR) is a binary up counter designed for monitoring program execution. If the program is operated into the unknown status by noise interference, WDC's overflow signal will reset this chip to restart operation.. The instruction that clear the watch-dog timer (BSET FWDRST) should be executed at proper points in a program within a given period. If an instruction that clears the watch-dog timer is not executed within the period and the watch-dog timer overflows, reset signal is generated and system is restarted with reset status. In order to generate different output timings, the user can control WDC by modifying WDrate bits of OSCM . This timer will be disabled at green and sleep modes.

OSCM initial value = xxx0 0x00

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	WTCKS	WDRST	WDrate	-	CPUM0	CLKMD	STPHX	HXWUP

WDrate : WDC's clock source select bit. 0 =14th, 1 =8th.
WDRST : Watch dog timer reset bit. 0 = Non reset, 1 = clear the watchdog timer's counter

An operation of watch-dog timer is as follows:

```
Main:
          B0BSET       FWDRST                ; Clear the watchdog timer's counter
          Call         sub1
          Call         sub2
          XXX
          Jmp          main
```

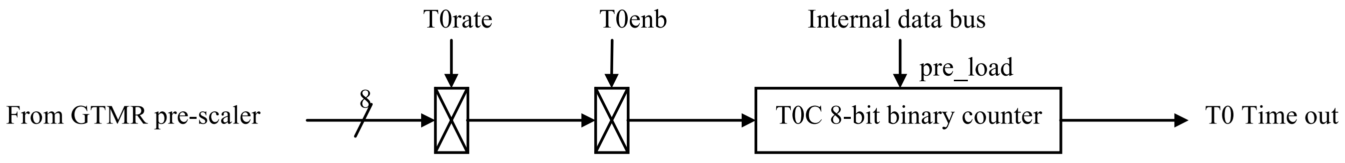
WDTMR counting time (High speed mode, fcpu = 3.58MHz / 4)

Wdrate	Watchdog overflow time
0	1 / (fcpu ÷ 2^{14} ÷ 16) = 293 ms
1	1 / (fcpu ÷ 2^{8} ÷ 16) = 4.5 ms
Watch-dog clock is internal RC low clock	1/(32 ÷ 16) = 0.5S @32Hz

Note : The watch dog timer can be enabled and be disabled by mask option.
The Wdrate can be set by the code option.

13. BASIC TIMER (T0)

The basic timer (T0) is an 8-bit binary up counter. It uses T0M register to select T0C's input clock for counting a precise time. If the T0 timer occurs an overflow (from FFH to 00H), it will continue counting and issue a time-out signal to trigger T0 interrupt to request interrupt service.



13.1 T0M MODE REGISTER

T0M initial value = 0xxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0M	T0ENB	T0RATE2	T0RATE1	T0RATE0	-	-	-	-

T0ENB : T0 timer controls bit. 0 = disable, 1 = enable.
T0RATE : The T0's clock source select bits. 000 = fcpu/256, 001 = fcpu/128, … , 110 = fcpu/4, 111 = fcpu/2.

13.2 T0C COUNTING REGISTER

T0C initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0C	x	X	x	x	x	x	x	x

The maximum interval time of T0 interrupt as follow

T0RATE	T0CLOCK	High speed mode (fcpu = 3.58MHz / 4)	
		Max overflow interval	One step = max/256
000	fcpu/256	73.2 ms	286us
001	fcpu/128	36.6 ms	143us
010	fcpu/64	18.3 ms	71.5us
011	fcpu/32	9.15 ms	35.8us
100	fcpu/16	4.57ms	17.9us
101	fcpu/8	2.28ms	8.94us
110	fcpu/4	1.14ms	4.47us
111	fcpu/2	0.57ms	2.23us

Note 1: The initial value of T0C register is calculated as follow

T0C initial value = 256 - (T0 interrupt interval time * input clock)

Note 2: The T0 timer must be disabled to modify T0C's value.

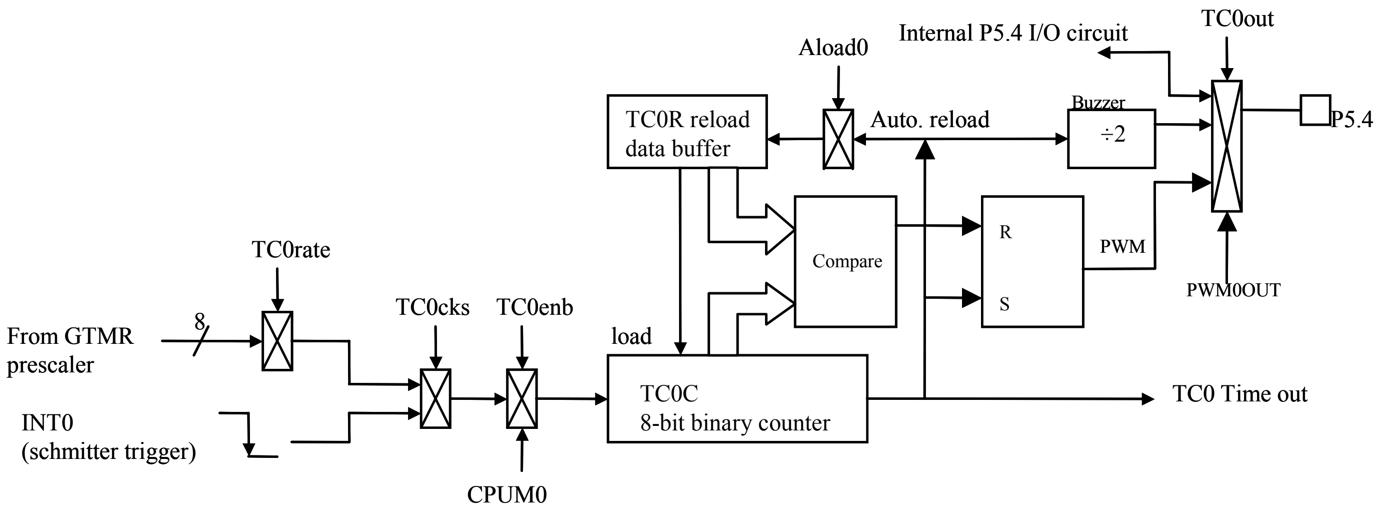
Example : To set 10ms interval time for T0 interrupt at 3.58MHz high speed mode. T0C value (74H) = 256 - (10ms * fcpu/64)

```
B0BCLR   FT0IEN        ; To disable T0 interrupt service
B0BCLR   FT0ENB        ; To disable T0 timer
MOV      A,#20H        ;
B0MOV    T0M,A         ; To set T0 clock = fcpu / 64
MOV      A,#74H
B0MOV    T0C,A         ; To set T0C initial value = 74H (To set T0 interval = 10 ms)
B0BSET   FT0IEN        ; To enable T0 interrupt service
B0BCLR   FT0IRQ        ; To clear T0 interrupt request
B0BSET   FT0ENB        ; To enable T0 timer
```

14. TIMER/EVENT COUNTER (TC0)

Timer/Event Counter 0 (TC0) is used to count system 'event' by identifying the transition (high-to-low) of incoming square wave signals. To indicate that an event has occurred, or that a specified time interval has elapsed, TC0 generates an interrupt request. By counting signal transitions and comparing the current counter value with the reference register value, TC0 can be used to measure specific time intervals.

TC0 has a auto re-loadable counter that consists of two parts: an 8-bit reload register (TC0R) into which you write the counter reference value, and an 8-bit counter register (TC0C) whose value is automatically incremented by counter logic.



TC0 Function Summary

8-bit programmable timer	Generates interrupts at specific time intervals based on the selected clock frequency.
External event counter	Count various system "events" based on edge detection of external clock signals at the P0.0 input pin.
Arbitrary frequency output	Outputs selectable clock frequencies to the TC0 output pin, TC0OUT
External signal divider	Divides the frequency of an incoming external clock signal according to a modifiable reference value (TC0R), and outputs the modified frequency to the TC0OUT
PWM function	PWM output can be generated by the PWM0OUT bit and output to TC0OUT.

14.1 TC0M MODE REGISTER

TC1M initial value = 0xxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0M	TC0ENB	TC0RATE2	TC0RATE1	TC0RATE0	TC0CKS	ALOAD0	TC0OUT	PWM0OUT

TC0ENB : TC0 counter/BUZZER0/PWM0OUT enable bit. 0 = disable, 1 = enable.
TC0RATE : TC0 internal clock select bits. 000 = fcpu/256, 001 = fcpu/128, … , 110 = fcpu/4, 111 = fcpu/2.
TC0CKS : TC0 clock source select bit. 0 = Internal clock source, 1 = External clock source (INTP0.0).
ALOAD : Auto-reload control bit. 0 = none auto-reload, 1 = auto-reload.
TC0OUT : TC0 time-out toggle signal output control bit.
0 = To disable TC0 signal output and to enable P5.4's I/O function,
1 = To enable TC0's signal output and to disable P5.4's I/O function. (Auto-disable the PWM0OUT fun.)
PWM0OUT : PWM output control bit
0 = To disable the PWM output
1 = To enable the PWM output (The TC0OUT control bit must = 0)

14.2 TC0C COUNTING REGISTER

TC1C initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0C	x	x	x	x	x	x	x	x

14.3 TC0R AUTO-LOAD REGISTER

TC1R initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0R	x	x	x	x	x	x	x	x

Note 1: The initial value of TC0C register is calculated as follows

TC0C initial value = 256 - (TC0 interrupt interval time * input clock)

Note 2: The TC0 timer must be disabled to modify TC0C's value.

14.4 PWM0 FUNCTION DESCRIPTION

The 8-bit counter counts modulus 256, that is, from 0-255, inclusive. The value of the 8-bit counter is compared to the contents of the reference register, RPWM0. When the reference register value equals the counter value, the PWM output goes low. When the counter reaches zero, the PWM output is forced high. The low-to-high ratio (duty) of the PWM output is RPWM0/256.

All PWM outputs remain inactive during the first 256 input clock signals. Then, when the counter value changes from FFH back to 00H, the PWM outputs are forced to high level. The pulse width ratio (duty cycle) is defined by the contents of the reference register and is programmed in increments of 1:256. The 8-bit PWM data register RPWM0 is read and written using 8-bit RAM control instruction only.

PWM output can be held at low level by continuously loading the reference register with 00H. By continuously loading the reference register with FFH, you can hold the PWM output to high level, except for the last pulse of the clock source, which sends the output low.

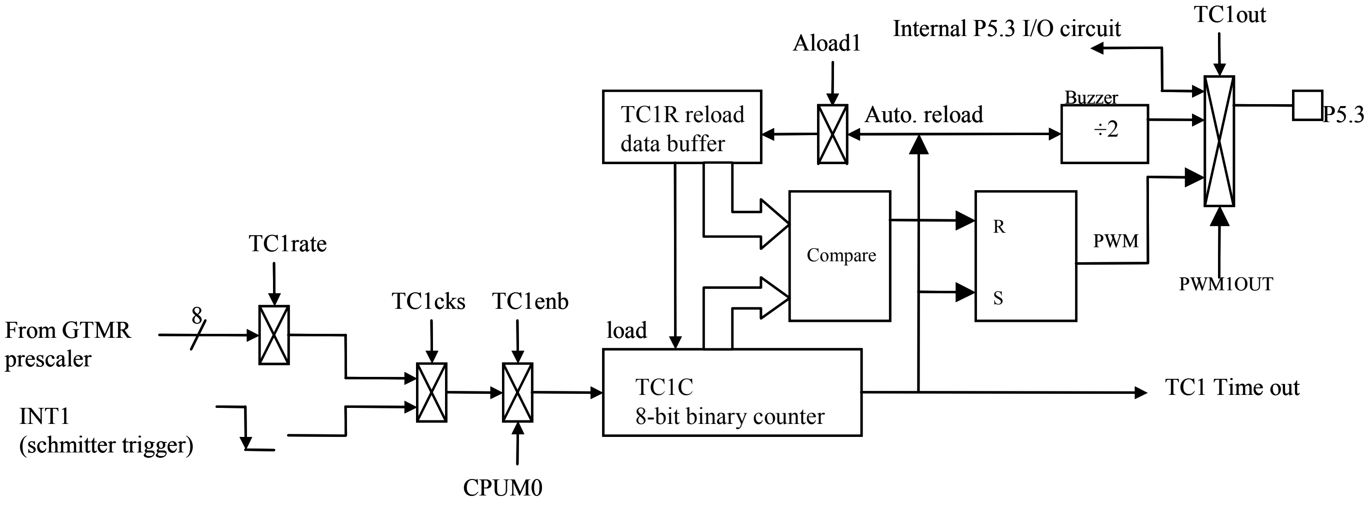
Reference Register Value (RPWM0)	Duty
0000 0000	0/256
0000 0001	1/256
0000 0010	2/256
---	---
---	---
1000 0000	128/256
1000 0001	129/256
---	---
---	---
1111 1110	254/256
1111 1111	255/256

Note : RPWM0 value is stored in TC0R register

15.TIMER/EVENT COUNTER (TC1)

Timer/Event Counter 1 (TC1) is used to count system 'event' by identifying the transition (high-to-low) of incoming square wave signals. To indicate that an event has occurred, or that a specified time interval has elapsed, TC1 generates an interrupt request. By counting signal transitions and comparing the current counter value with the reference register value, TC1 can be used to measure specific time intervals.

TC1 has a auto re-loadable counter that consists of two parts: an 8-bit reload register (TC1R) into which you write the counter reference value, and an 8-bit counter register (TC1C) whose value is automatically incremented by counter logic.



TC1 Function Summary

8-bit programmable timer	Generates interrupts at specific time intervals based on the selected clock frequency.
External event counter	Count various system "events" based on edge detection of external clock signals at the P0.1 input pin.
Arbitrary frequency output	Outputs selectable clock frequencies to the TC1 output pin, TC1OUT
External signal divider	Divides the frequency of an incoming external clock signal according to a modifiable reference value (TC1R), and outputs the modified frequency to the TC1OUT
PWM function	PWM output can be generated by the PWM1OUT bit and output to TC1OUT.

15.1 TC1M MODE REGISTER

TC1M initial value = 0xxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1M	TC1ENB	TC1RATE2	TC1RATE1	TC1RATE0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT

TC1ENB : TC1 counter/BUZZER1/PWM1OUT enable bit. 0 = disable, 1 = enable.
TC1RATE : TC1 internal clock select bits. 000 = fcpu/256, 001 = fcpu/128, … , 110 = fcpu/4, 111 = fcpu/2.
TC1CKS : TC1 clock source select bit. 0 = Internal clock source, 1 = External clock source (INTP0.1).
ALOAD : Auto-reload control bit. 0 = none auto-reload, 1 = auto-reload.
TC1OUT : TC1 time-out toggle signal output control bit.
0 = To disable TC1 signal output and to enable P5.3's I/O function,
1 = To enable TC1's signal output and to disable P5.3's I/O function. (Auto-disable the PWM1OUT fun.)
PWM1OUT : PWM output control bit
0 = To disable the PWM output
1 = To enable the PWM output (The TC1OUT control bit must = 0)

15.2 TC1C COUNTING REGISTER

TC1C initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1C	x	x	x	x	x	x	x	x

15.3 TC1R AUTO-LOAD REGISTER

TC1R initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1R	x	x	x	x	x	x	x	x

Note 1: The initial value of TC1C register is calculated as follows

TC1C initial value = 256 - (TC1 interrupt interval time * input clock)

Note 2: The TC1 timer must be disabled to modify TC1C's value.

15.4 PWM1 FUNCTION DESCRIPTION

The 8-bit counter counts modulus 256, that is, from 0-255, inclusive. The value of the 8-bit counter is compared to the contents of the reference register, RPWM1. When the reference register value equals the counter value, the PWM output goes low. When the counter reaches zero, the PWM output is forced high. The low-to-high ratio (duty) of the PWM output is RPWM1/256.

All PWM outputs remain inactive during the first 256 input clock signals. Then, when the counter value changes from FFH back to 00H, the PWM outputs are forced to high level. The pulse width ratio (duty cycle) is defined by the contents of the reference register and is programmed in increments of 1:256. The 8-bit PWM data register RPWM1 is read and written using 8-bit RAM control instruction only.

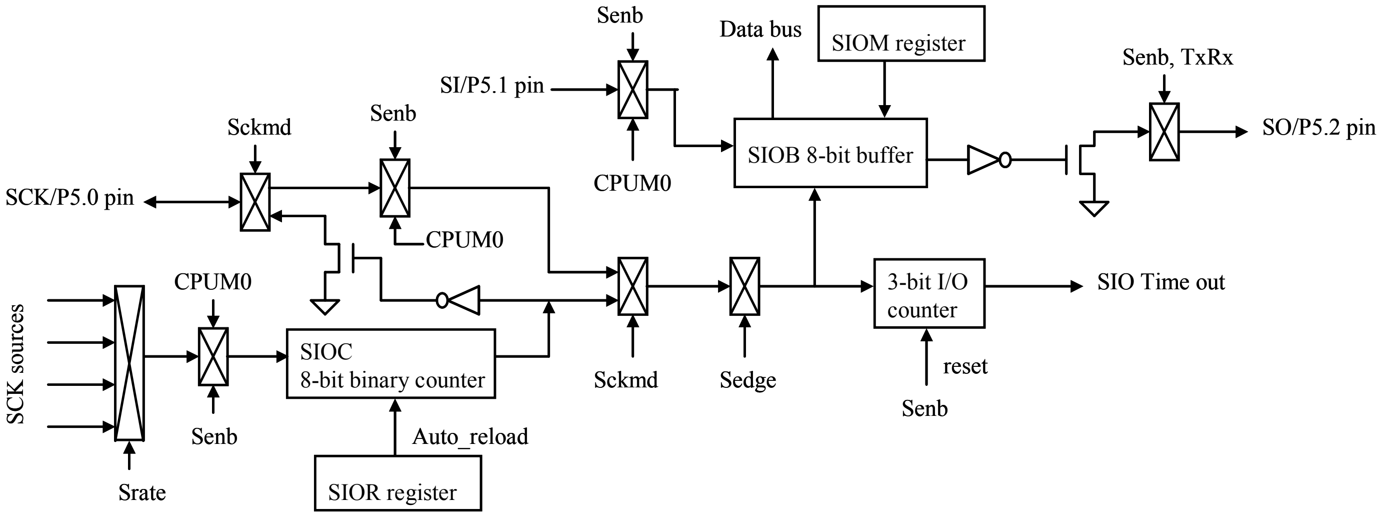
PWM output can be held at low level by continuously loading the reference register with 00H. By continuously loading the reference register with FFH, you can hold the PWM output to high level, except for the last pulse of the clock source, which sends the output low.

Reference Register Value (RPWM1)	Duty
0000 0000	0/256
0000 0001	1/256
0000 0010	2/256
---	---
---	---
1000 0000	128/256
1000 0001	129/256
---	---
---	---
1111 1110	254/256
1111 1111	255/256

Note : RPWM1 value is stored in TC1R register

16. SERIAL INPUT/OUTPUT TRANSCEIVER (SIO)

The HE8P1700A provides an 8-bit SIO interface circuit with clock rate selection. The SIOM register can control SIO operating function, such as: transmit/receive, clock rate, transfer edge and starting this circuit. This SIO circuit will automatic TX or RX 8-bit data by setting SENB and START bits in SIOM register. The SIOB is an 8-bit buffer, which is designed to store transfer data. SIOC and SIOR is designed to generate SIO's clock source with auto-reload function. The 3-bit I/O counter can monitor the operation of SIO and announce an interrupt request after transmitting/receiving 8 bits data. After transferring 8-bit data, this circuit will be disabled automatically and re-transfer data by programming SIOM register.



16.1 SIOM MODE REGISTER

SIOM initial value = 0xxx -xxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SIOM	SENB	START	SRATE1	SRATE0	-	SCKMD	SEDGE	TXRX

SENB : SIO function control bit. 0 = disable (P5.0~P5.2 is general purpose port), 1 = enable (P5.0~P5.2 is SIO pins).
START : SIO progress control bit. 0 = End of transfer, 1 = progressing.
SRATE1,0 : SIO's clock select bit. 00 = fcpu, 01 = fcpu/32, 10 = fcpu/16, 11 = fcpu/8.
SCKMD : SIO's clock mode select bit. 0 = internal, 1 = external mode.
SEDGE : SIO's transfer clock edge select bit. 0 = falling edge, 1 = raising edge.
TXRX : SIO's transfer direction select bit. 0 = receiver only , 1 = transmitter/receiver full duplex

16.2 SIOB/SIOR DATA BUFFER

SIOB/SIOR initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SIOB/SIOR	x	X	x	x	x	x	x	x

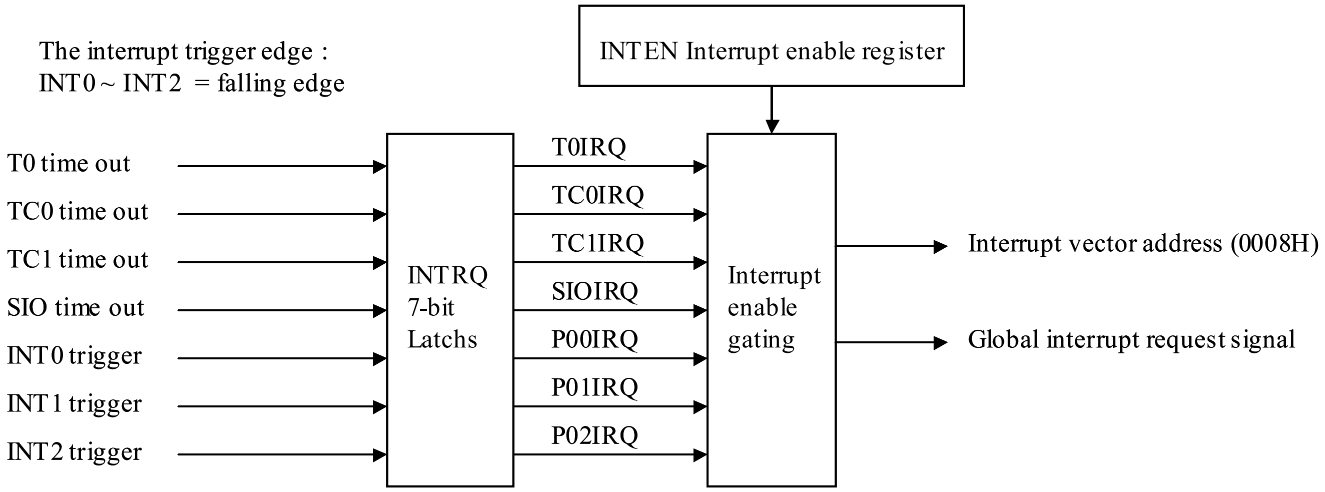
SIOB and SIOR are 8-bit data buffers. The SIOR is designed for SIOC to reload the counted value when end of counting.

Note : To setup SIOR's value for getting desire transfer clock as follows

SIOR value = 256 - (desired SIO's clock * selected input clock)

17. INTERRUPT

The HE8P1700A provides 7 interrupt sources, including four internal interrupts (T0, TC0, TC1 & SIO) and three external interrupts (INT0 ~ INT2). These external interrupts can warm up the chip while the system is switched from power-down to high-speed mode. The external clock input pins of TC0/TC1 are shared with P0.0/P0.1 pins. Beside this, all of the Port 0 and Port 1 pins can work with warm up function. Once interrupt service is executed, the GIE bit in STKP register will clear to "0" for stopping other interrupt request. On the contrast, when interrupt service exits, this bit will set to "1" to accept the next interrupts' request. All of the interrupt request signals are stored in INTRQ register. The user can program the chip to check INTRQ's content for setting executive priority.



17.1 INTEN INTERRUPT ENABLE REGISTER

INTEN initial value = x000 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTEN	-	TC1IEN	TC0IEN	T0IEN	SIOIEN	P02IEN	P01IEN	P00IEN

P00IEN : External P0.0 interrupt control bit. 0 = disable, 1 = enable.
P01IEN : External P0.1 interrupt control bit. 0 = disable, 1 = enable.
P02IEN : External P0.2 interrupt control bit. 0 = disable, 1 = enable.
SIOIEN : SIO interrupt control bit. 0 = disable, 1 = enable.
T0IEN : T0 timer interrupt control bit. 0 = disable, 1 = enable.
TC0IEN : Timer/event counter 0 interrupt control bit. 0 = disable, 1 = enable.
TC1IEN : Timer/event counter 1 interrupt control bit. 0 = disable, 1 = enable.

17.2 INTRQ INTERRUPT REQUEST REGISTER

INTRQ initial value = x0x0 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTRQ	-	TC1IRQ	TC0IRQ	T0IRQ	SIOIRQ	P02IRQ	P01IRQ	P00IRQ

P00IRQ : External P0.0 interrupt request bit. 0 = non-request, 1 = request.
P01IRQ : External P0.1 interrupt request bit. 0 = non-request, 1 = request.
P02IRQ : External P0.2 interrupt request bit. 0 = non-request, 1 = request.
SIOIRQ : SIO interrupt request bit. 0 = non-request, 1 = request.
T0IRQ : T0 timer interrupt request control bit. 0 = non request, 1 = request.
TC0IRQ : TC0 timer/event counter interrupt request controls bit. 0 = non request, 1 = request.
TC1IRQ : TC1 timer/event counter interrupt request controls bit. 0 = non request, 1 = request.

Example : Interrupt service routine

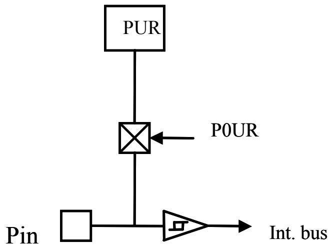
```
              ORG       0008H                    ;
INTRS:        PUSH                               ; To push working registers.
              B0XCH     A,ACCBUF                 ; To push ACC.
              B0BTS0    FP00IRQ                  ; To skip, if P0.0 did not have interrupt request.
              JMP       P00INTR                  ;
              B0BTS0    FT0IRQ                   ; To skip, if T0 did not have interrupt request.
              JMP       T0INTR                   ;
              .         .                        ;
QINTRS:       B0MOV     A,ACCBUF                 ; To pop ACC.
              POP                                ; To pop working registers.
              RETI      .                        ; To exit interrupt routine.
              .         .                        ;

P00INTR:      B0BCLR    FP00IRQ                  ; To clear P0.0IRQ bit for waiting next time request.
              .         .                        ;
              JMP       QINTRS                   ;
;                       .
T0INTR:       B0BCLR    FT0IRQ                   ; To clear T0 interrupt request bit for waiting next time request.
              .         .                        ;
              JMP       QINTRS                   ;
```

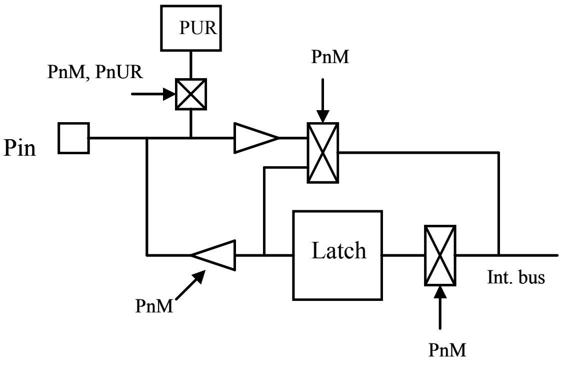
18. I/O PORT

The HE8P1700A provides 5 ports for users' application, consisting of one input port (P0) and four I/O ports (P1, P2, P4, P5). The input pull up resistor of each pin can be programmed by PnUR register. The direction of I/O port is selected by PnM register. After the system resets, these ports work as input function without pull up resistor. If the user want to read-in a signal from I/O pin, it recommends to switch I/O pin as input mode and to execute read-in instruction. (B0BTS0 M.b, B0BTS1 M.b or B0MOV A,M).

Port0 structure



Port1, 2, 4, 5 structure



Note : All of the latch output circuits are push-pull structures.

18.1 PORT 1 WAKEUP (P1W) REGISTER

In the power down mode or green mode, one of port 1 pin has a logic "L" signal, it can wakeup this chip into normal mode operation. In this case, the P1.n pin must be set to input mode by P1M control and its wakeup function is programmed by P1W register.

P1W initial value = xx00 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1W	-	-	P15W	P14W	P13W	P12W	P11W	P10W

P1nW : Port 1.n wakeup control bit. 0 = without wakeup function, 1 = with wakeup function.

18.2 PULL-UP RESISTOR (PnUR) REGISTER

PnUR initial value = 0000 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PnUR	Pn7R	Pn6R	Pn5R	Pn4R	Pn3R	Pn2R	Pn1R	Pn0R

Pn : The n expressed 0, 1, 2, 4 and 5.
Pn7R ~ Pn0R : pull-up resistor control bit. 0 = without pull up resistor, 1 = with pull up resistor.

18.3 PORT MODE (PnM) REGISTER

PnM initial value = 0000 0000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PnM	Pn7M	Pn6M	Pn5M	Pn4M	Pn3M	Pn2M	Pn1M	Pn0M

Pn.M : The n expressed 1, 2, 4 and 5.
Pn7M ~ Pn0M : Port n.7 ~ Port n.0 input/output mode control bit. 0 = input, 1 = output.

18.4 PORT (Pn) DATA REGISTER

Pn initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pn : The n expressed 0, 1, 2, 4 and 5.
Pn7 ~ Pn0 : Port n.7 ~ Port n.0 input/output data bit. 0 = logic “Low”, 1 = Logic “High”.

19. 8-CHANNEL ANALOG TO DIGITAL CONVERTER

This analog converter uses 8-input sources with up to 256-step resolution to transfer analog signal into 8-bits digital data. The sequence of ADC operation is to select input source (AIN0 ~ AIN7) at first and then set GCHS and ADS bit to “1” to start conversion. After twelve comparative steps are taken, the ADC circuit will set EOC bit to “1” and final value output in ADB register. This ADC circuit can select 8-bit resolution operation by programming ADLEN bit in ADR register.

19.1 ADM REGISTER

ADM initial value = 0xx0 xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADM	ADENB	ADS	EOC	GCHS	-	CHS2	CHS1	CHS0

CHS2,1,0 : 8 channels select bit. 000 = AIN0, 001 = AIN1, 010 = AIN2, 011 = AIN3 , .. , 111 = AIN7.
GCHS : Global channel select bit. 0 = To disable AIN channel, 1 = To enable AIN channel.
EOC : ADC status bit. 0 = Progressing, 1 = End of converting and reset ADENB bit.
ADS : ADC start bit. 0 = stop, 1 = starting.
ADENB : ADC control bit. 0 = disable, 1 = enable.

19.2 ADB & ADR REGISTERS

ADB initial value = xxxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

ADR initial value = xxx0 xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADR	-	ADCKS	ADLEN	TADC	ADB3	ADB2	ADB1	ADB0

ADBn : ADC data buffer. ADB11~ADB4 bits for 8-bit ADC.
TADC : ADC test mode control bit. 0 = normal mode, 1 = test mode.
ADLEN : ADC’s resolution select bits. 0 = 8-bit.
ADCKS : ADC’s clock source select bit. 0 = fosc/2, 1 = fosc.

Example : To set AIN0 ~ AIN1 for ADC input and executing 8-bit ADC

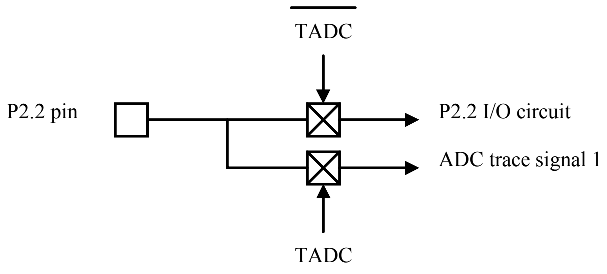
```
ADC0:       B0BCLR  FTADC              ; To enable ADC circuit & to disable test mode
            B0BCLR  FADLEN             ; To set 8-bit ADC operation
            MOV     A,#90H
            B0MOV   ADM,A              ; To enable ADC and set AIN0 input
            B0BSET  FADS               ; To start conversion
WADC0:      B0BTS1  FEOC               ; To skip, if end of converting =1
            JMP     WADC0              ;   else, jump to WADC0
            B0MOV   A,ADB              ; To get AIN0 input data
ADC1:       MOV     A,#91H             ;
            B0MOV   ADM,A              ; To enable ADC and set AIN1 input
            B0BSET  FADS               ; To start conversion
            .       .                  .
QEXADC:     B0BCLR  FGCHS              ; To release AINx input channel
```

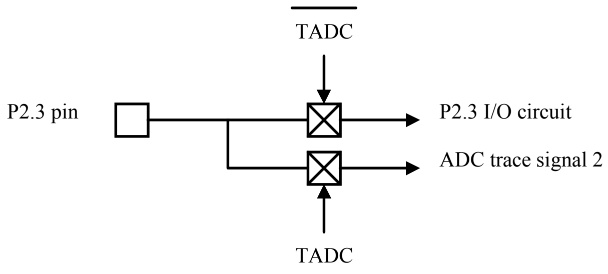
The AIN's input voltage v.s. ADB's output data

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
1/4096*AVREF	0	0	0	0	0	0	0	0	0	0	0	0
2/4096*AVREF	0	0	0	0	0	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
4094/4096*AVREF	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*AVREF	1	1	1	1	1	1	1	1	1	1	1	1

19.3 ADC TEST MODE

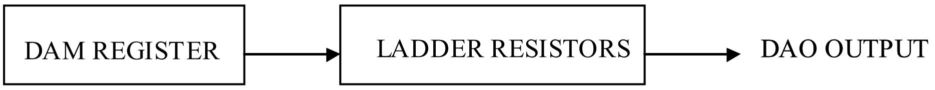
The P2.2 & P2.3 pins can be used to testing ADC circuit performance. When TADC bit sets to logic "H", then P2.2 & P2.3 pins work for tracing ADC's signals. Otherwise, it work as general purpose I/O pin.





20. 7-BIT DIGITAL TO ANALOG CONVERTER

The D/A converter uses 7-bit structure to synthesize 128 steps' analog signal with current source output. After DAENB bit is set to "1", DAC circuit will turn to be enabled and the DAM register, from bit0 to bit6, will send digital signal to ladder resistors in order to generate analog signal on DAO pin.



DAM initial value = 0xxx xxxx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DAM	DAENB	DAB6	DAB5	DAB4	DAB3	DAB2	DAB1	DAB0

DABn : Digital input data.
DAENB : Digital to Analog converter control bit, 0 = disable, 1 = enable.

The DAB's data v.s. DAO's output voltage as follow

DAB6	DAB5	DAB4	DAB3	DAB2	DAB1	DAB0	DAO
0	0	0	0	0	0	0	VSS
0	0	0	0	0	0	1	Idac
0	0	0	0	0	1	0	2 * Idac
0	0	0	0	0	1	1	3 * Idac
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1	1	1	1	1	1	0	126 * Idac
1	1	1	1	1	1	1	127 * Idac

21. ABSOLUTE MAXIMUM RATING

(All of the voltages referenced to Vss)

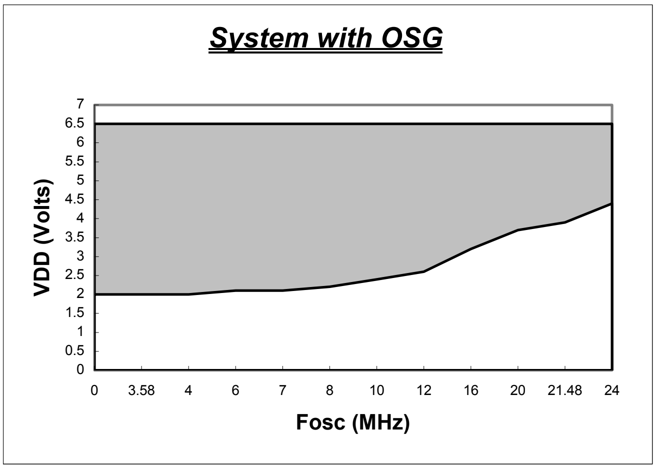
Supply voltage (Vdd)	- 0.3V ~ 6.0V
Input in voltage (Vin)	Vss - 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr)	0°C ~ + 70°C
Storage ambient temperature (Tstor)	–30°C ~ + 125°C
Power consumption (Pc)	500 mW

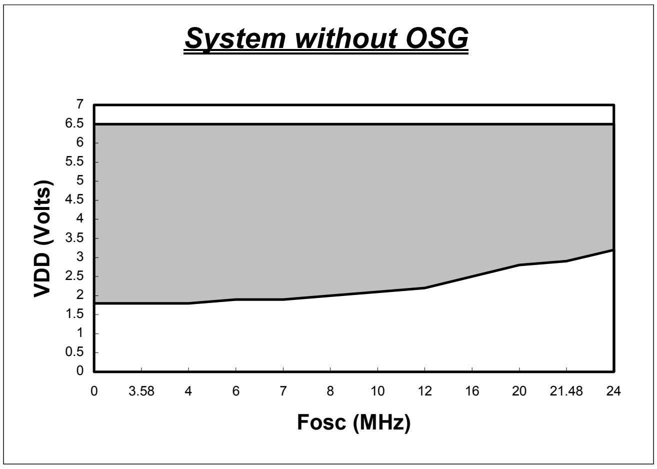
22.ELECTRICAL CHARACTERISTIC

(All of voltages referenced to Vss, Vdd = 5.0V, fosc = 3.579545 MHz, ambient temperature is 25℃ unless otherwise note.)

PARAMETER	SYM.	DESCRIPTION		MIN.	TYP.	MAX.	UNIT
Operating voltage	Vdd	Normal mode, Vpp = Vdd		2.4	5.0	5.5	V
		Programming mode, Vpp = 12.5V		4.5	5.0	5.5	
RAM Data Retention voltage	Vdr			-	1.5	-	V
Internal POR	Vpor	Vdd rise rate to ensure internal power-on reset		-	0.05	-	V/ms
Input Low Voltage	ViL1	All input pins except those specified below		Vss	-	0.3Vdd	V
	ViL2	Input with Schmitt trigger buffer - Port0		Vss	-	0.2Vdd	V
	ViL3	Reset pin ; Xin (in RC mode)		Vss	-	0.2Vdd	V
	ViL4	Xin (in X'tal mode)		Vss	-	0.3Vdd	V
Input High Voltage	ViH1	All input pins except those specified below		0.7Vdd	-	Vdd	V
	ViH2	Input with Schmitt trigger buffer –Port0		0.8Vdd	-	Vdd	V
	ViH3	Reset pin ; Xin (in RC mode)		0.9Vdd	-	Vdd	V
	ViH4	Xin (in X'tal mode)		0.7Vdd	-	Vdd	V
Reset pin leakage current	Ilekg	Vin = Vdd		-	-	2	uA
I/P port pull-up resistor	Rup	Vin = Vss , Vdd = 5V		-	100	-	KΩ
I/P port input leakage current	Ilekg	Pull-up resistor disable, Vin = Vdd		-	-	2	uA
Port1 output source current	IoH	Vop = Vdd - 0.5V		-	15	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	15	-	
Port2 output source current	IoH	Vop = Vdd - 0.5V		-	15	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	15	-	
Port4 output source current	IoH	Vop = Vdd - 0.5V		-	15	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	15	-	
Port5 output source current	IoH	Vop = Vdd - 0.5V		-	15	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	15	-	
INTn trigger pulse width	Tint0	INT0 ~ INT2 interrupt request pulse width		2/fcpu	-	-	cycle
AVREF input voltage	Varef	Vdd = 5.0V		2	-	Vdd	V
AIN0 ~ AIN7 input voltage	Vani			Vss+0.2	-	Avref	V
DA output Current	IDAO	DAM = $7F @ Vout=0.7V , Vdd = 3V		5	10	-	mA
Supply Current	Idd1	Run Mode	Vdd= 5V 4Mhz	-	5	8.5	mA
			Vdd= 3V 4Mhz	-	1.5	3	mA
			Vdd= 3V 32768Hz	-	50	90	uA
	Idd2	Internal RC mode (16KHz)	Vdd= 5V	-	18	40	uA
			Vdd= 3V	-	15	30	uA
	Idd3	Stop mode	Vdd= 5V	-	9	15	uA
			Vdd= 3V	-	2.5	6	uA
Voltage detector current	Ivdet	LVD enable operating current		-	100	180	uA

22.1HE8P1700A SERIES OTP OSCILLATOR FREQUENCY vs. OPERATING VOLTAGE GRAPH


System with OSG


System without OSG

X'tal (Hz)	OSG		No OSG		Cxin (p)	Cxout(p)
	min. (V)	max. (V)	min. (V)	max. (V)		
3.58M	2.0	6.5	1.8	6.5	20	20
4M	2.0	6.5	1.8	6.5	20	20
6M	2.1	6.5	1.8	6.5	20	20
7M	2.1	6.5	1.9	6.5	20	20
8M	2.2	6.5	2.0	6.5	20	20
10M	2.4	6.5	2.1	6.5	20	20
12M	2.6	6.5	2.2	6.5	20	20
16M	3.2	6.5	2.5	6.5	20	20
20M	3.7	6.5	2.8	6.5	20	20
21.48M	3.9	6.5	2.9	6.5	20	20
24M	4.4	6.5	3.2	6.5	20	20

Note : 1. The shaded region indicate the permissible combinations of VDD and Fosc.

2. Please adjust capacitances of Xin and Xout in OSG status. It will change the minimum voltage of working frequency.

ex. Fosc = 20MHz, VDD = 3V, Cxin = 47p, Cxout = 10p.

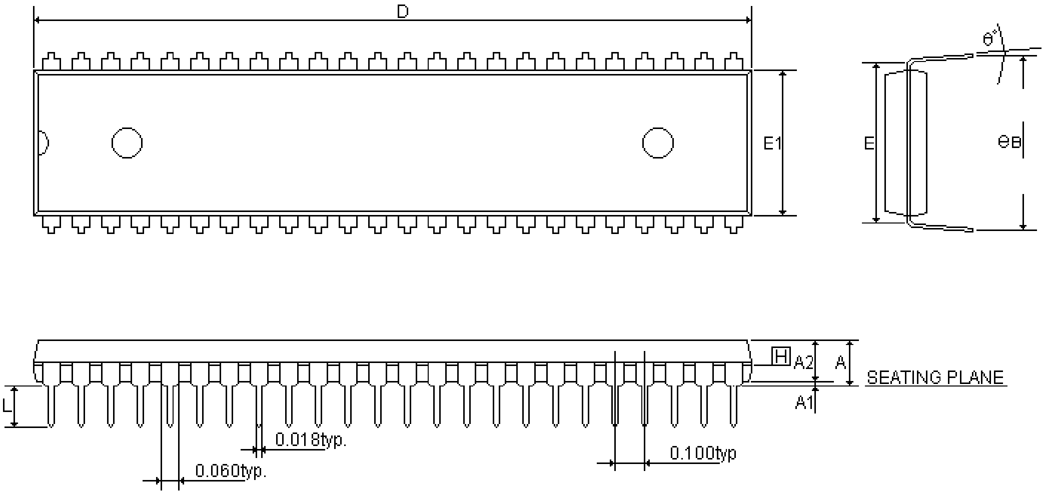
23.INSTRUCTION SET

Field	Mnemonic		Description	C	DC	Z	Cycle
MOVE	MOV	A,M	A ← M	-	-	√	1
	MOV	M,A	M ← A	-	-	-	1
	B0MOV	A,M	A ← M (bnak 0)	-	-	√	1
	B0MOV	M,A	M (bank 0) ← A	-	-	-	1
	MOV	A,I	A ← I	-	-	-	1
	B0MOV	M,I	M ← I, (M = only for Working registers R, Y, Z , RBANK & PFLAG)	-	-	-	1
	XCH	A,M	A ←→M	-	-	-	1
	B0XCH	A,M	A ←→M (bank 0)	-	-	-	1
	MOVC		R, A ← ROM [Y,Z]	-	-	-	2
ARITHMETIC	ADC	A,M	A ← A + M + C, if occur carry, then C=1, else C=0	√	√	√	1
	ADC	M,A	M ← A + M + C, if occur carry, then C=1, else C=0	√	√	√	1
	ADD	A,M	A ← A + M, if occur carry, then C=1, else C=0	√	√	√	1
	ADD	M,A	M ← M + A, if occur carry, then C=1, else C=0	√	√	√	1
	B0ADD	M,A	M (bank 0) ← M (bank 0) + A, if occur carry, then C=1, else C=0	√	√	√	1
	ADD	A,I	A ← A + I, if occur carry, then C=1, else C=0	√	√	√	1
	SBC	A,M	A ← A - M - /C, if occur borrow, then C=0, else C=1	√	√	√	1
	SBC	M,A	M ← A - M - /C, if occur borrow, then C=0, else C=1	√	√	√	1
	SUB	A,M	A ← A - M, if occur borrow, then C=0, else C=1	√	√	√	1
	SUB	M,A	M ← A - M, if occur borrow, then C=0, else C=1	√	√	√	1
	SUB	A,I	A ← A - I, if occur borrow, then C=0, else C=1	√	√	√	1
	DAA		To adjust ACC's data format from HEX to DEC.	√	-	-	1
	MUL	A,M	R, A ← A * M, The LB of product stored in Acc and HB stored in R register. ZF affected by Acc.	-	-	√	2
LOGIC	AND	A,M	A ← A and M	-	-	√	1
	AND	M,A	M ← A and M	-	-	√	1
	AND	A,I	A ← A and I	-	-	√	1
	OR	A,M	A ← A or M	-	-	√	1
	OR	M,A	M ← A or M	-	-	√	1
	OR	A,I	A ← A or I	-	-	√	1
	XOR	A,M	A ← A xor M	-	-	√	1
	XOR	M,A	M ← A xor M	-	-	√	1
	XOR	A,I	A ← A xor I	-	-	√	1
PROCESS	SWAP	M	A (b3~b0, b7~b4) ←M(b7~b4, b3~b0)	-	-	-	1
	SWAPM	M	M(b3~b0, b7~b4) ← M(b7~b4, b3~b0)	-	-	-	1
	RRC	M	A ← RRC M	√	-	-	1
	RRCM	M	M ← RRC M	√	-	-	1
	RLC	M	A ← RLC M	√	-	-	1
	RLCM	M	M ← RLC M	√	-	-	1
	CLR	M	M ← 0	-	-	-	1
	BCLR	M.b	M.b ← 0	-	-	-	1
	BSET	M.b	M.b ← 1	-	-	-	1
	B0BCLR	M.b	M(bank 0).b ← 0	-	-	-	1
	B0BSET	M.b	M(bank 0).b ← 1	-	-	-	1
BRANCH	CMPRS	A,I	ZF,C ← A - I, If A = I, then skip next instruction	√	-	√	1 + S
	CMPRS	A,M	ZF,C ← A – M, If A = M, then skip next instruction	√	-	√	1 + S
	INCS	M	A ← M + 1, If A = 0, then skip next instruction	-	-	-	1 + S
	INCMS	M	M ← M + 1, If M = 0, then skip next instruction	-	-	-	1 + S
	DECS	M	A ← M - 1, If A = 0, then skip next instruction	-	-	-	1 + S
	DECMS	M	M ← M - 1, If M = 0, then skip next instruction	-	-	-	1 + S
	BTS0	M.b	If M.b = 0, then skip next instruction	-	-	-	1 + S
	BTS1	M.b	If M.b = 1, then skip next instruction	-	-	-	1 + S
	B0BTS0	M.b	If M(bank 0).b = 0, then skip next instruction	-	-	-	1 + S
	B0BTS1	M.b	If M(bank 0).b = 1, then skip next instruction	-	-	-	1 + S
	JMP	d	PC15/14 ← RomPages1/0, PC13~PC0 ← d	-	-	-	2
	CALL	d	Stack ← PC15~PC0, PC15/14 ← RomPages1/0, PC13~PC0 ← d	-	-	-	2
MISC	RET		PC ← Stack	-	-	-	2
	RETI		PC ← Stack, and to enable global interrupt	-	-	-	2
	PUSH		To push working registers (080H~087H) into buffers	-	-	-	1
	POP		To pop working registers (080H~087H) from buffers	√	√	√	1
	NOP		No operation	-	-	-	1

Note : a). Working registers = R, Y and Z. b). The memory is access to location RAM[Y,Z], if M = @YZ (located at address E7H in RAM bank 0). c). The memory is access to location RAM[Y,Z], if M = @YZ (located at address 0E7H in RAM bank 0). d). All instructions are one cycle operation except for program branch and PC update which are two cycles.

24.PACKAGE INFORMATION

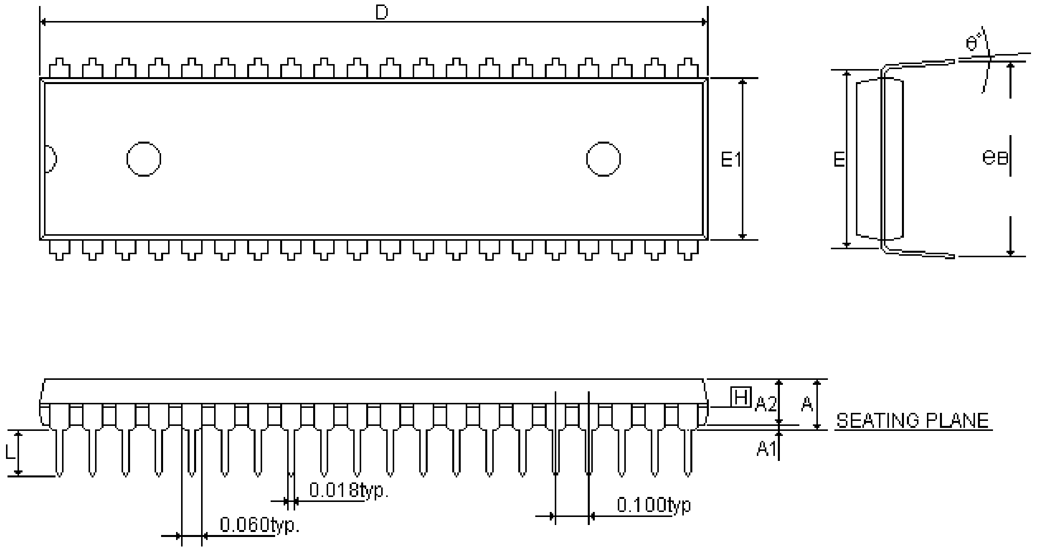
P-DIP48 pin :



Symbols	MIN.	NOR.	MAX.
A	-	-	0.220
A1	0.015	-	-
A2	0.150	0.155	0.160
D	2.400	2.450	2.550
E	0.600 BSC		
E1	0.540	0.545	0.550
L	0.115	0.130	0.150
eB	0.630	0.650	0.670
θ°	0	7	15

UNIT : INCH

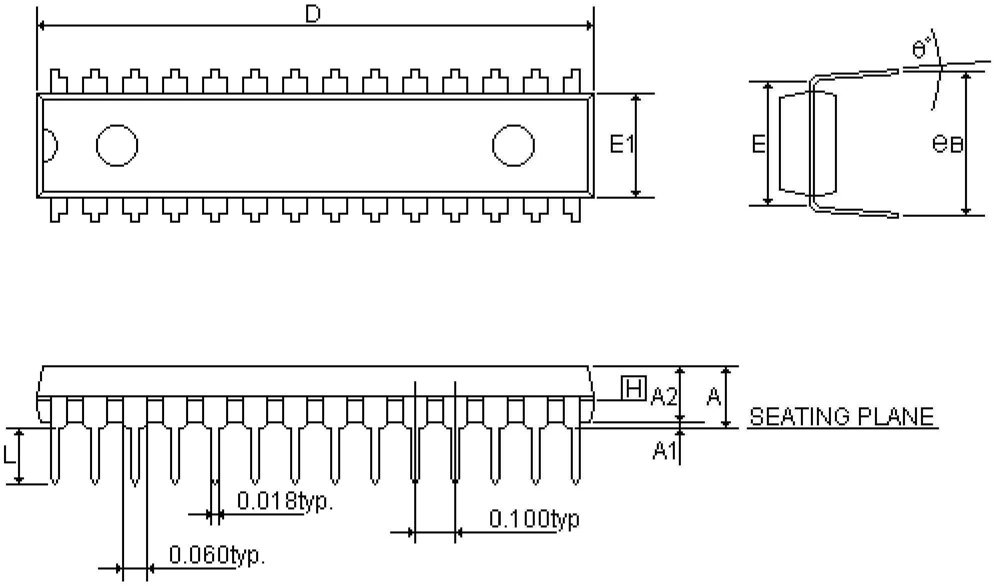
P-DIP40 pin :



Symbols	MIN.	NOR.	MAX.
A	-	-	0.220
A1	0.015	-	-
A2	0.150	0.115	0.160
D	2.055	2.060	2.070
E	0.600BSC.		
E1	0.540	0.545	0.550
L	0.115	0.130	0.150
e_B	0.630	0.650	0.670
θ°	0	7	15

UNIT : INCH

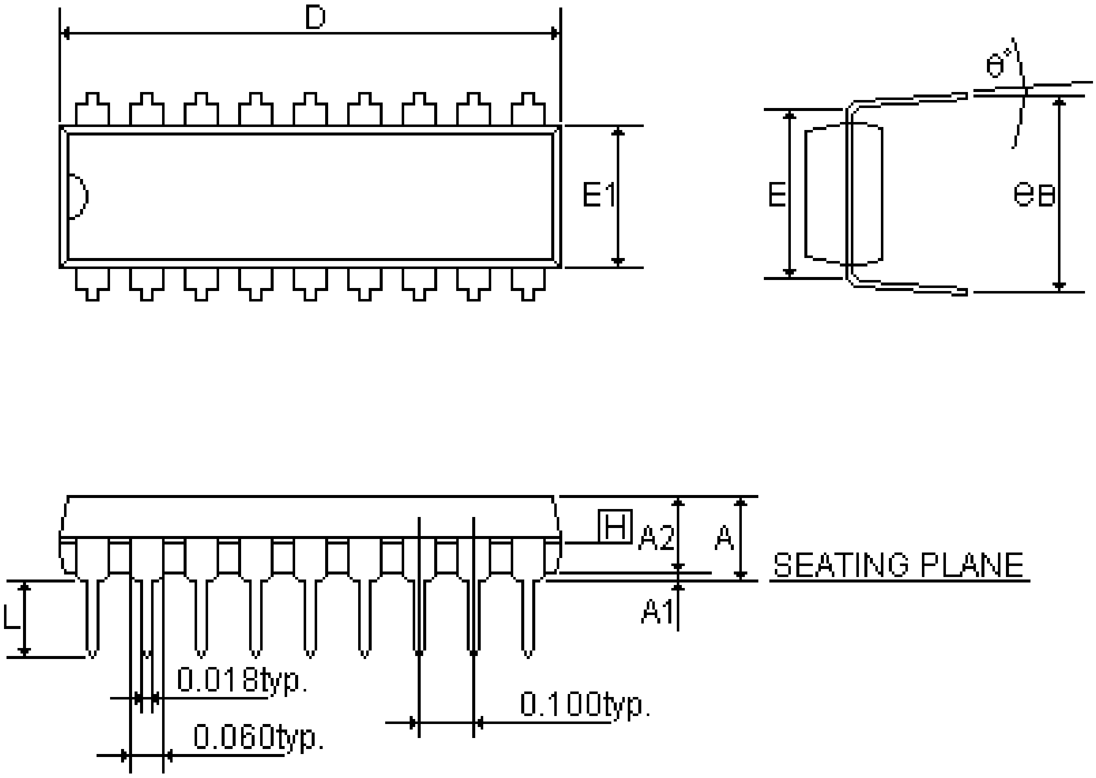
S-DIP28 pin :



Symbols	MIN.	NOR.	MAX.
A	-	-	0.210
A1	0.015	-	-
A2	0.114	0.130	0.135
D	1.390	1.390	1.400
E	0.310BSC.		
E1	0.283	0.288	0.293
L	0.115	0.130	0.150
e_B	0.330	0.350	0.370
θ°	0	7	15

UNIT : INCH

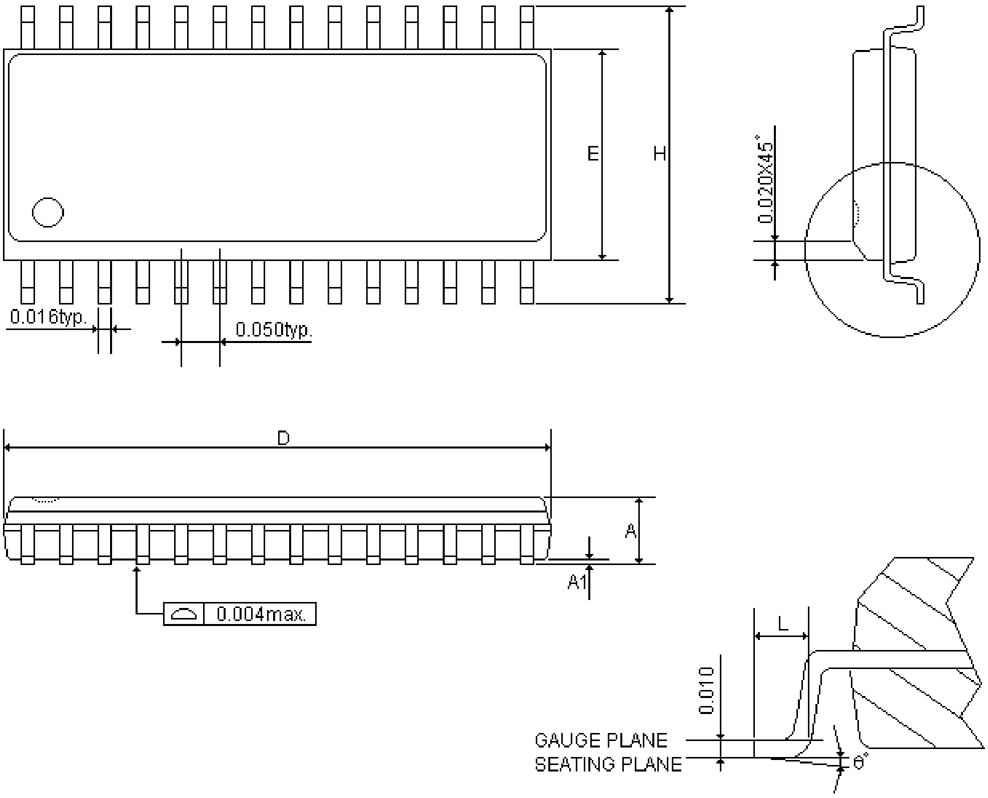
P-DIP18 pin :



Symbols	MIN.	NOR.	MAX.
A	-	-	0.210
A1	0.015	-	-
A2	0.125	0.130	0.135
D	0.880	0.900	0.920
E	0.300BSC.		
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
e B	0.335	0.355	0.375
θ°	0	7	15

UNIT : INCH

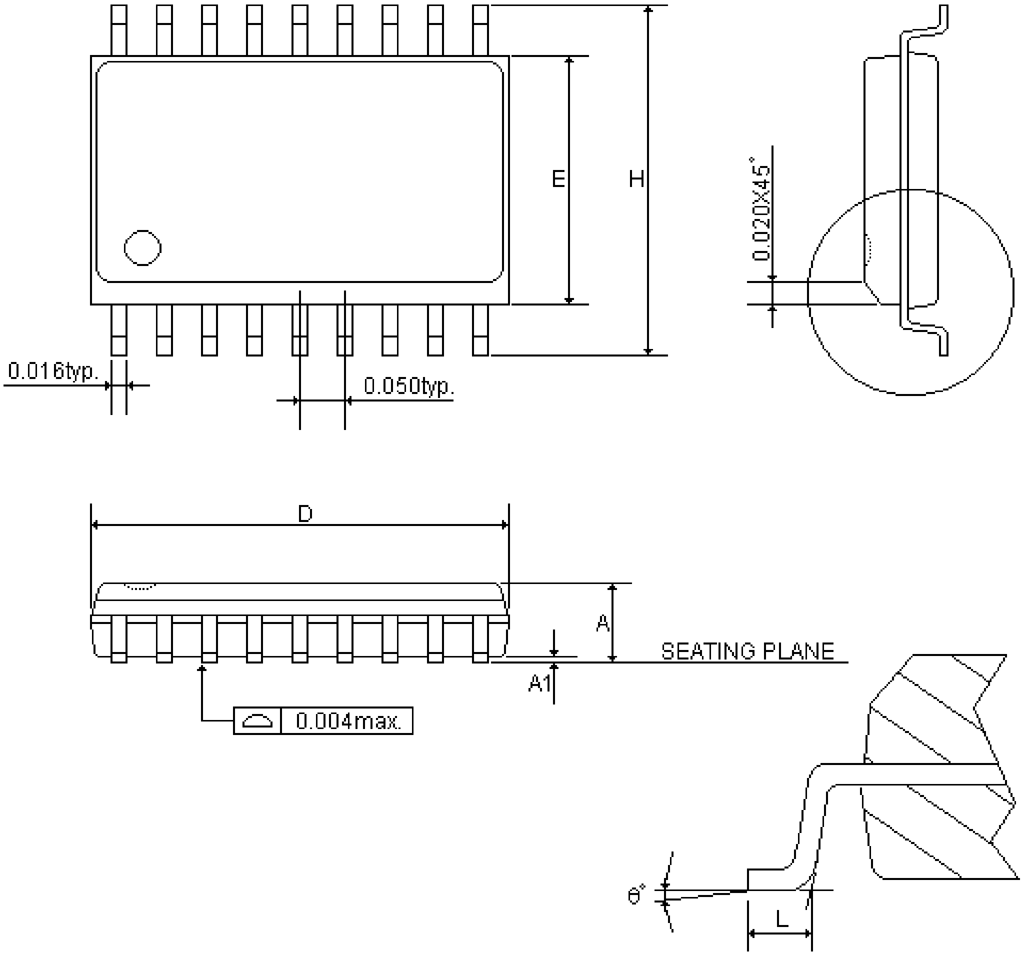
SOP28 pin :



Symbols	MIN.	MAX.
A	0.093	0.104
A1	0.004	0.012
D	0.697	0.713
E	0.291	0.299
H	0.394	0.419
L	0.016	0.050
θ°	0	8

UNIT : INCH

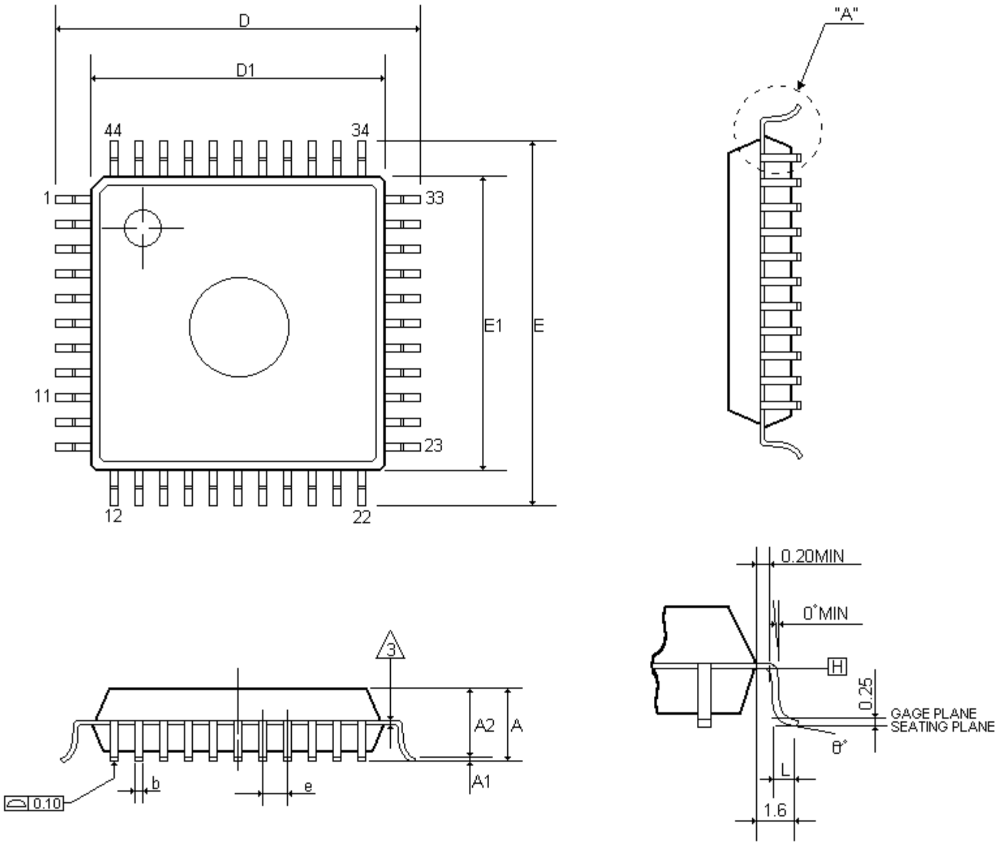
SOP18 pin:



Symbols	MIN.	MAX.
A	0.093	0.104
A1	0.004	0.012
D	0.447	0.463
E	0.291	0.299
H	0.394	0.419
L	0.016	0.050
θ°	0	8

UNIT : INCH

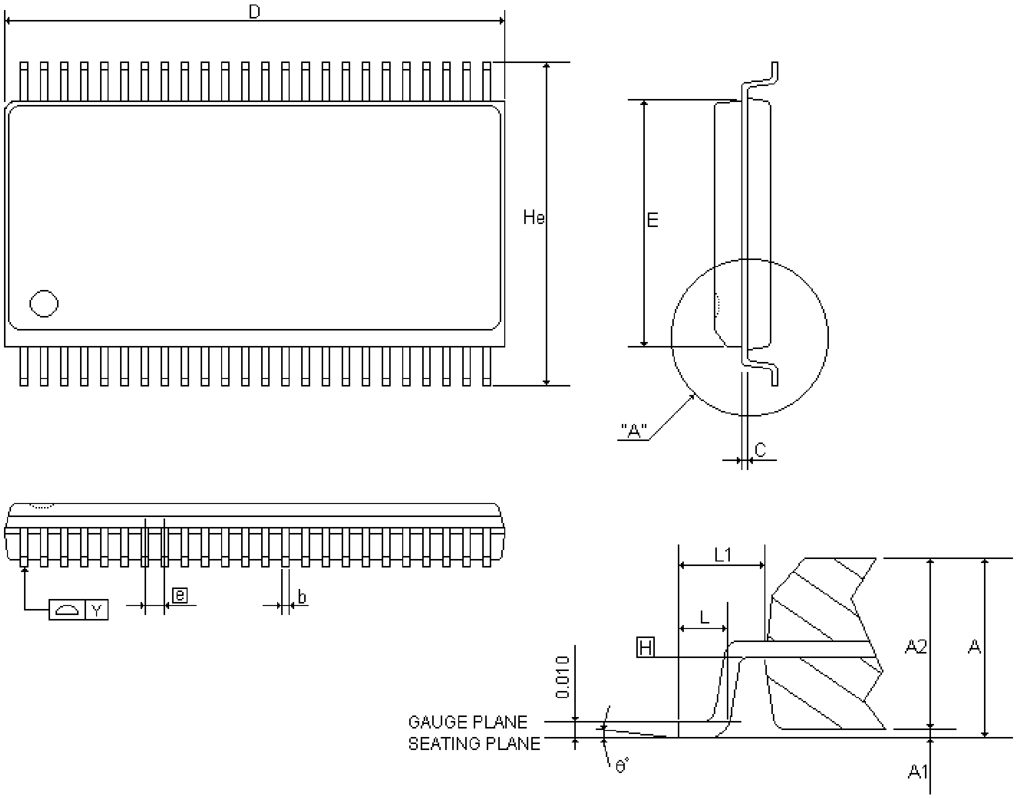
QFP44 pin :



Symbols	MIN.	NOR.	MAX.
A	-	-	2.7
A1	0.25	0.30	0.35
A2	1.9	2.0	2.2
b	0.3 (TYP.)		
D	13.00	13.20	13.40
D1	9.9	10.00	10.10
E	13.00	13.20	13.40
E1	9.9	10.00	10.10
L	0.73	0.88	0.93
e	0.8 (TYP.)		
θ°	0	-	7
C	0.1	0.15	0.2

UNIT : mm

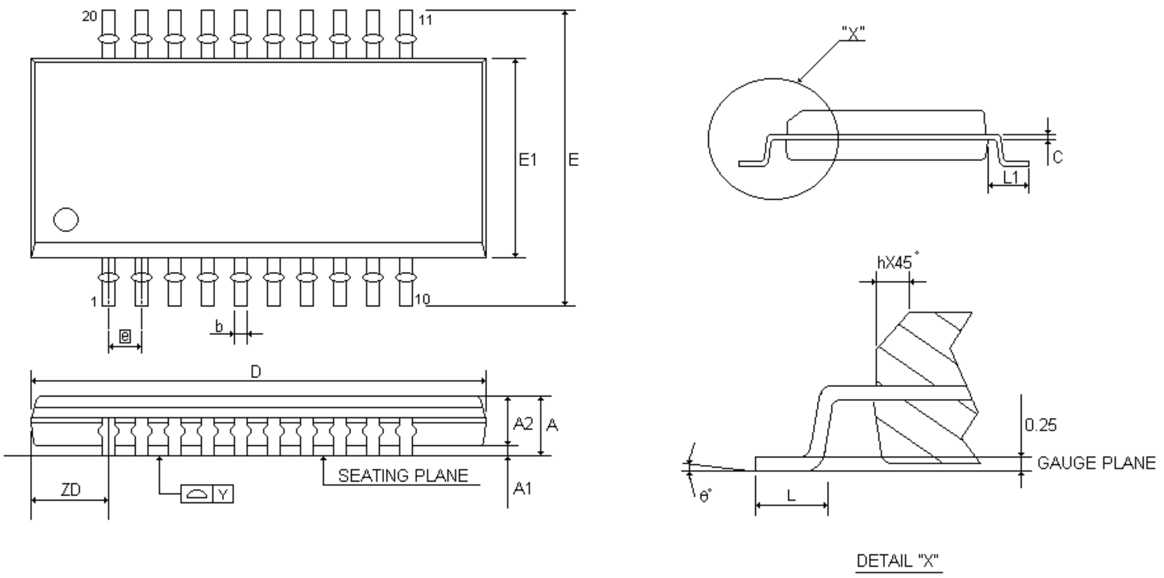
SSOP48 pin :



Symbols	MIN.	NOR.	MAX.
A	0.095	0.102	0.110
A1	0.008	0.012	0.016
A2	0.089	0.094	0.099
b	0.008	0.010	0.013
C	-	0.008	-
D	0.620	0.625	0.630
E	0.291	0.295	0.299
H	-	0.025	-
He	0.396	0.406	0.416
L	0.020	0.030	0.040
L1	-	0.056	-
Y	-	-	0.003
θ°	0	-	8

UNIT : INCH

SSOP20 pin



Symbols	DIMENSION (MM)			DIMENSION (MIL)		
	MIN,	NOM.	MAX.	MIN.	NOM.	MAX.
A	1.35	1.60	1.75	53	63	69
A1	0.10	0.15	0.25	4	6	10
A2	-	-	1.50	-	-	59
b	0.20	0.254	0.30	8	10	12
b1	0.20	0.254	0.28	8	11	11
C	0.18	0.203	0.25	7	8	10
C1	0.18	0.203	0.23	7	8	9
D	8.56	8.66	8.74	337	341	344
E	5.80	6.00	6.20	228	236	244
E1	3.80	3.90	4.00	150	154	157
e	0.635 BSC			25 BSC		
h	0.25	0.42	0.50	10	17	20
L	0.40	0.635	1.27	16	25	50
L1	1.00	1.05	1.10	39	41	43
ZD	1.50 REF			58 REF		
Y	-	-	0.10	-	-	4
θ°	0°	-	8°	0°	-	8°