CST4A0XXA/B Series Speech Flash

1. CST4A0XXA/B General Descriptions

CST4A0XXA/B series are 4bit micro-controller which could play 4 channel melody or 4 channel ADPCM with PWM direct drive circuit. PWM resolution is 12 bits. They includes a low cost, high performance CMOS micro-processor. The clock frequency of this up is typically 13.1072 (±3%) MHz. This chip operates over a wide voltage range of 1.8V~5.5V. It contains program ROM and data ROM inside. The maximum program ROM is 8K word and data ROM size is 32K ~ 124K byte depending on PROM density. The maximum working SRAM is (512+2) nibbles. It provides with total 4 software programmable I/O Ports.

2. CST4A0XXA/B Features

- Operating voltage: 1.8V to 5.5V
- Internal HROSC: 65.536 MHz, LRCOSC 32K Hz
  - MCU Operation frequency: 13.1072 MHz
- Memory Size
  - Flash Program Memory capacity: 2KX12, 4KX12, 6KX12 or 8KX12 bits selected by option.
  - Flash Data ROM capacity: 32Kx8 ~ 124KX8 bits depending on PROM density.
  - SRAM size: 512X4 bits
  - User register: 2X4 bits
- Wakeup function for power-down mode
  - HALT mode wakeup source: Port A, PWM and RTC wakeup can wake-up from HALT mode to NORMAL mode and executing wake-up sub-routine program.
- 4 input/output pins: Port A can be defined as input or output port bit by bit.
- Three reset condition
  - Low voltage reset (LVR = 1.7V)
  - Power on reset
  - Watch dog timer overflow
- One internal interrupt sources
  - PWM interrupt
- WDT
  - Watch dog timer, can enabled/disabled by option
  - WDT period is 0.23 sec (Clock source by LRCOSC = 32.000K Hz ± 50%).
- Audio output
  - Support 8/10/12 bits PWM
- Support option set for pull down resistor 1M/(50K or 220K), low voltage reset…etc.
- Oscillator fuse option ±3%, temperature & voltage compensation
- Support security option (1 bit) to protect user code.
- Low voltage detect (LVD)
- 12 sign x 8 unsigned multiplication
- Support indirect jump instruction JMPX, JMPPMA.
- Support direct call instruction CALLX, CALLPMA.
- Up to 8 level stacks for call sub-routing.
- TOUCH function: 4 I/O
- WKRTC
  - Real-time clock in halt mode, can enabled/disabled by option
  - wakeup period is 16/32/64/128 ms select by user code.
  - Only operating on Halt mode.
    (Clock source by LRCOSC = 32.000K Hz± 50%)

CST4A0XXA/B Series Speech Flash

2.1 CSTF4A0XXA/B series selection table

Part No.	DROM Size	IO Keep High with halt mode	IO Keep High with Deep Sleep
CST4A010A	32Kx8 ~ 44Kx8		V
CST4A020A	64Kx8 ~ 76Kx8		V
CST4A030A	96Kx8 ~ 108Kx8		V
CST4A040A	112Kx8 ~ 124Kx8		V
CST4A010B	32Kx8 ~ 44Kx8	V	V
CST4A020B	64Kx8 ~ 76Kx8	V	V
CST4A030B	96Kx8 ~ 108Kx8	V	V
CST4A040B	112Kx8 ~ 124Kx8	V	V

Table 1. Selection table

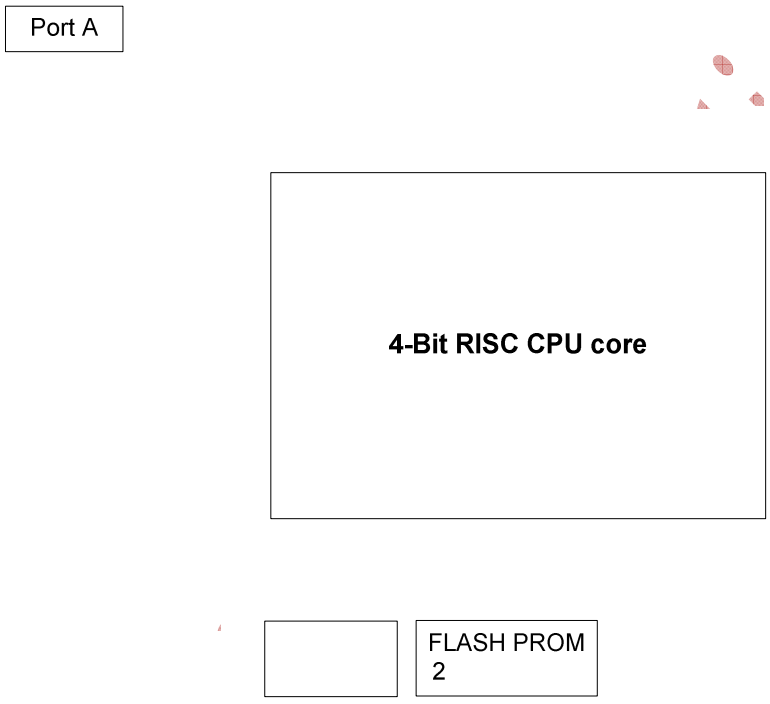
3. Packaging and Pads Information

3.1 Pads

PAD Name	Type	State After Reset	Description
Reset, Power Input			
VCC	P	High	Power input of I/O port
VSS	P	Low	Ground input except PWM block power. It could be double bonded with VPS pad.
General I/O ports			
PA0~PA3	I/O	XXXX	Port A is a programmable Input /Output port.
Audio output pads			
PWMP	O	Low	Audio output PWM(+) .
PWMN	O	Low	Audio output PWM(-) .

Table 2. Pad Description

3.2 Block Diagram



3.3 Package

CST4A0XXA/B series provides SOP8

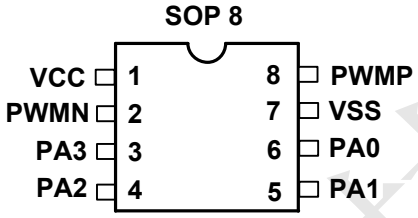


Figure 2. SOP8 Package

4.3 AC Characteristics

VCC=3.0V, Ta=25℃ unless otherwise noted

Parameters	Symbol	Minimum	Typical	Maximum
Operating Frequency (HROSC/5)	Fsys	12.714MHz	13.1072MHz	13.500MHz
LRCOSC	Flrc	16K Hz	32K Hz	48K Hz
RC reset time-constant	Trrc	-	10 us	-
Program ROM data ready time	Tprr	-	-	2/Fsys
Data ROM data ready time	Tdrr	-	-	2/Fsys

Table 5. AC Characteristics

5. FUNCTIONAL DESCRIPTION

This chapter describes the function of MCU.

5.1 Flash Program Memory (PROM)

The flash program memory has a capacity of 2/4/6/8K X12-bit (0000H ~ 1FFFH). The last 256 location of effective PROM is reserved area for testing program. The user shall not use this area in any case. Assembler shall check user program on this limit. Hardware does not need to check this restriction.
There is one option “OTPREV” to reserve unused area of PROM. If all unused area of PROM need to fill with “0XFFF”, the option “OTPREV” on the IDE tool must be enabled. Otherwise, they will fill with “0x000”.

	PROM 23	PROM 43	PROM (3	PROM 83
user area	0x000~0x6EF	0x000~0xEEF	0x0000~0x16EF	0x0000~0x1EEF
Reserve	0x6F0~0x6FB	0xEF0~0xEFB	0x16F0~0x16FB	0x1EF0~0x1EFB
serial number	0x6FC~0x6FF	0xEFC~0xEFF	0x16FC~0x16FF	0x1EFC~0x1EFF
system testing area	0x700~0x7FF	0xF00~0xFFF	0x1700~0x17FF	0x1F00~0x1FFF

Table 6. Memory Map of PROM

5.2 Flash Data ROM (DROM)

The Flash Data ROM has a capacity of 32KX8 ~ 124KX8 bit, It stores the 8 bits wide data for ADPCM or melody data …etc. The last 64 location is a reserved area. The user shall not use this area in any case. Assembler shall check user data ROM on this limit. Hardware does not need to check this restriction.

<table>
<tr><td rowspan="2">DROM Addess</td><td colspan="4">CST4A010A</td></tr>
<tr><td>PROM 8K</td><td>PROM 6K</td><td>PROM 4K</td><td>PROM 2K</td></tr>
<tr><td>0000 ~ 0FFF</td><td rowspan="3">user area</td><td rowspan="5">user area</td><td rowspan="7">user area</td><td rowspan="9">user area</td></tr>
<tr><td>…</td></tr>
<tr><td>7F00 ~ 7FBF</td></tr>
<tr><td>7FC0 ~ 7FFF</td><td>system area, last 64 location (don’t use)</td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>8FC0 ~ 8FFF</td><td></td><td>system area, last 64 location (don’t use)</td></tr>
<tr><td>…</td><td></td><td></td><td></td></tr>
<tr><td>9FC0 ~ 9FFF</td><td></td><td></td><td>system area, last 64 location (don’t use)</td></tr>
<tr><td>…</td><td></td><td></td><td></td></tr>
<tr><td>AFC0 ~ AFFF</td><td></td><td></td><td></td><td>system area, last 64 location (don’t use)</td></tr>
</table>

Table 7. CST4A010A Memory Map of DROM

CST4A0XXA/B Series Speech Flash

DROM Addess	CST4A020A			
	PROM 8K	PROM 6K	PROM 4K	PROM 2K
00000 ~ 00FFF	user area	user area	user area	user area
01000 ~ 01FFF				
…				
0FF00 ~ 0FFBF				
0FFC0 ~ 0FFFF	system area, last 64 location (don't use)			
…				
10FC0 ~ 10FFF		system area, last 64 location (don't use)		
…				
11FC0 ~ 11FFF			system area, last 64 location (don't use)	
…				
12FC0 ~ 12FFF				system area, last 64 location (don't use)

Table 8. CST4A020A Memory Map of DROM

DROM Addess	CST4A030A			
	PROM 8K	PROM 6K	PROM 4K	PROM 2K
00000 ~ 00FFF	user area	user area	user area	user area
01000 ~ 01FFF				
…				
0FF00 ~ 0FFBF				
17FC0 ~ 17FFF	system area, last 64 location (don't use)			
…				
18FC0 ~ 18FFF		system area, last 64 location (don't use)		
…				
19FC0 ~ 19FFF			system area, last 64 location (don't use)	
…				
1AFC0 ~ 1AFFF				system area, last 64 location (don't use)

Table 9. CST4A030A Memory Map of DROM

CST4A0XXA/B Series Speech Flash

DROM Addess	CST4A040A			
	PROM 8K	PROM 6K	PROM 4K	PROM 2K
00000 ~ 00FFF	user area	user area	user area	user area
01000 ~ 01FFF				
…				
0FF00 ~ 0FFBF				
1BFC0 ~ 1BFFF	system area, last 64 location (don't use)			
…				
1CFC0 ~ 1CFFF		system area, last 64 location (don't use)		
…				
1DFC0 ~ 1DFFF			system area, last 64 location (don't use)	
…				
1EFC0 ~ 1EFFF				system area, last 64 location (don't use)

Table 10. CST4A040A Memory Map of DROM

DROM is addressed by five registers DMA4, DMA3, DMA2, DMA1 and DMA0. After these registers are specified by software, the data need enough delay time, Tdrr in the table of "AC Characteristics". After this delay time, the data can be read from data register (DMDL & DMDH). Otherwise, the data you read will be unknown.

DMA0	18H	R/W	xxxx	DMA0.3	DMA0.2	DMA0.1	DMA0.0	DMA0~4, five register built a 18 bit addressing space for read DROM 8 bits data, DMA0 is lowest nibble, DMA4 is highest nibble of DROM address
DMA1	19H	R/W	xxxx	DMA1.3	DMA1.2	DMA1.1	DMA1.0	
DMA2	1AH	R/W	xxxx	DMA2.3	DMA2.2	DMA2.1	DMA2.0	
DMA3	1BH	R/W	xxxx	DMA3.3	DMA3.2	DMA3.1	DMA3.0	
DMDL	1CH	R/W	xxxx	DMDL.3	DMDL.2	DMDL.1	DMDL.0	Low nibble of DROM data read from this address . Write this register means DMA add one.
DMDH	1DH	R/W	xxxx	DMDH.3	DMDH.2	DMDH.1	DMDH.0	High nibble of DROM data read from this address
DMA4	1EH	R/W	--xx	-	-	-	DMA4.0	DMA4 is highest nibble of DROM address DMA4[3:1] is not implemented.

Table 11. SFRs about DROM

Extended Common registers :

WFLASHEN	**0BH**	**R/W**	0000	**WDEN**	**EDEN**	0	0	**WDEN :** 0: Don't care. Auto cleared by hardware. 1: Flash DROM writing start. **EDEN :** 0: Don't care. Auto cleared by hardware. 1: Flash DROM erasing start.
WEDROMCTL	**0CH**	**R/W**	0000	**0**	**0**	**INT3EN**	**INT3FG**	**INT3EN:** 0: Write or Erase DROM int. disabled. 1: Write or Erase DROM int. enabled. **INT3FG:** 0: No erasing or writing DROM has been finished. 1: It will be set high after erasing or writing DROM has been finished.

Table 12. Flash DROM operation register

Reading Data from the Flash DROM

DROM is addressed by five registers DMA4, DMA3, DMA2, DMA1 and DMA0. After these registers are specified by software, the data can be read from DMDL and DMDH register.

Writing Data to the Flash DROM

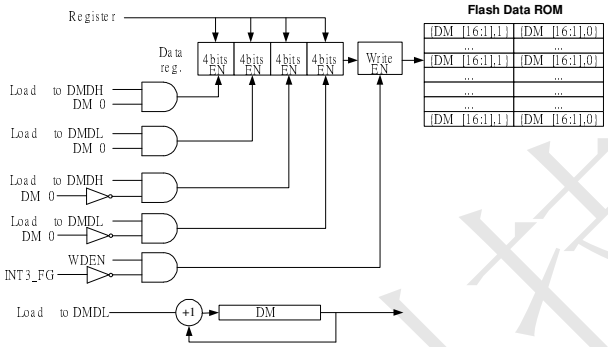


Figure 3. Write Flash DATA DROM block diagram

Notice:

When erasing or writing Flash DROM, the DROM address assigned by DMA4~DMA0 registers must be legal address. Otherwise, PROM data may be corrupted.

The width of Flash DROM is 8 bits, but it's really implement by 16 bit width. So, to write two bytes data(16 bits) to Flash DROM in one time is must. A 16-bit Write Buffer is used to store these two bytes data temporarily and it can be stored by writing data to DMDH and DMDL twice. Single byte write isn't supported on this device. Note that writing the 16 bit Writer Buffer must follow the sequence of writing DMDH first, and then DMDL, DMDH and DMDL register. It's not recommended to change the DMDH and DMDL writing sequence mention above. The DROM's address (DMA4~DMA0) have a hardware adder to mange it, and it will increase by one after writing DMDL register automatically. Base on physically Flash DROM structure, to initial a write cycle the address of DROM should be an even number.

The Flash DROM address of the data to be written must first be placed in the DMA4~DMA0 registers, DMDH and DMDL registers. To write data to the Flash DROM, the write enable bit WDEN must be set to high to enable the write function. The WDEN register will be cleared to low after set high automatically by hardware. Writing cycle takes about Twr (47us)time to write a data to DROM after WDEN set high, the INT3FG flag will be set high after writing cycle finished. During writing cycle, MCU will not execute any instruction until INT3FG set high. Programmer must clear INT3FG to "0" before next erasing or writing. The WDEN set high is not allowed by hardware when INT3FG keeps high.

Flash DROM structure

Sector	DMA4~DMA0 Address		R/W function
Sector 0 (512 bytes)	0x00001(8 bit)	0x00000(8bit)	Erasable/Writable by application program
	…	…	
	0x001FF(8 bit)	0x001FE(8 bit)	
Sector 1 (512 bytes)	0x00201(8 bit)	0x00200(8 bit)	
	…	…	
	0x003FF	0x003FE	
…	…	…	
	…	…	
Sector 248 (512 bytes)	0x1EE01	0x1EE00	
	…	…	
	0x1EFFF	0x1EFFE	

Table 13. Memory Map of DROM

For example, write Data to DROM Address 0x01000 ~0x01001

```
    LD      A, #0H
    LD      (DMA4), A
    LD      A, #0H
    LD      (DMA3), A
    LD      A, #FH
    LD      (DMA2), A
    LD      A, #FH
    LD      (DMA1), A
    LD      A, #EH
    LD      (DMA0), A       // set DMA address 0x00FFE
    LDPCH WRITE_DROM
    CALL   WRITE_DROM
    …
    …

WRITE_DROM:

    LD      A, (20H)
    LD      (DMDH), A       // store high nibble to reg. for 0x01000
    LD      A, (21H)
    LD      (DMDL), A       // store low nibble to reg. for 0x01000, then DMA+1, DMA = 0x00FFF
    LD      A, (22H)
    LD      (DMDH), A       // store high nibble to reg. for 0x01001
    LD      A, (23H)
    LD      (DMDL), A       // store low nibble to reg. for 0x01001 , then DMA+1, DMA = 0x01000

    LD      A, #0000B
    LD      EXIO(WEDROMFG), A
    LD      A, #1000B
    LD      EXIO(WEDROMEN), A  // write Data from reg. to FLASH, DMA = 0x01001 and 0x01000

WRFG_LOOP:
    LD      A, EXIO(WEDROMEN)
    AND     A, #1000B
    CMP     A, #1000B
    JZ      WRFG_LOOP
    LD      A, EXIO(WEDROMFG)
    AND     A, #0001B
    CMP     A, #0001B
    JNZ     WRFG_LOOP

    RETS
```

Erasing Data on the Flash DROM

Flash DROM Data can be erased by user program on a sector-by-sector basis. The sector architecture is based on sector size of 512 Bytes. There is only one the necessary sector will be erased, and the sector is assigned by DMA4~DMA0 register.

The Flash DROM address of the data to be erased must first be placed in the DMA4~DMA0 registers. To erase data on the Flash DROM, the erase enable bit EDEN must be set to high to enable the erase function. The EDEN register will be cleared to low after set high automatically by hardware.

Easing cycle takes about Ter (3ms) time to erase a sector on DROM after EDEN set high, the INT3FG flag will be set high after erasing cycle finished. During erasing cycle, MCU will not execute any instruction until

INT3FG set high. Programmer must clear INT3FG to “0” before next erasing or writing. The EDEN set high is not allowed by hardware when INT3FG keeps high.

For example, Erase DROM Address 0x01000 ~0x011ff

```
        LD      A, #0H
        LD      (DMA4), A
        LD      A, #1H
        LD      (DMA3), A
        LD      A, #0H
        LD      (DMA2), A
        LD      A, #0H
        LD      (DMA1), A
        LD      A, #0H
        LD      (DMA0), A       // set DMA address 0x01000

        LDPCH ERASE_DROM
        CALL  ERASE_DROM
        ....
        ....
    ERASE_DROM:
        LD      A, #0000B
        LD      EXIO(WEDROMFG), A
        LD      A, #0100B
        LD      EXIO(WEDROMEN), A

    ERFG_LOOP:
        LD      A, EXIO(WEDROMEN)
        AND     A, #0100B
        CMP     A, #0100B
        JZ      ERFG_LOOP
        LD      A, EXIO(WEDROMFG)
        AND     A, #0001B
        CMP     A, #0001B
        JNZ     ERFG_LOOP
        RETS
```

5.3 SRAM and Special Function Register

There are two kinds of SRAM addressing mode, describe below:

1. **Direct addressing mode:**
   There are 512 nibbles SRAM in this chip. The SRAM and I/O memory map is divided into several pages by setting MAH register (3 bits wide). The initial value of MAH is unknown and must be defined by instruction “LDMAH” before you utilize SRAM. The extra 2 SRAM nibbles in the specifications and hardware manuals of relative mask ROM products are not SRAM in fact. They are USER1 and USER2 of SFRs.

Direct Addressing		SRAM MAP
MAH=XH	00H~1FH	SFR(special function register) register
MAH=0H	20H~3FH	USER SRAM
MAH=1H	20H~3FH	
~	20H~3FH	
MAH=15H	20H~3FH	

Table 14. Memory Map of SFRs

The first 32-nibble, 00H ~ 1FH, are defined as a common block. Some I/O and user register is arranged in this common block for easy data operations. The other regions, 20H~3FH, are employed as SRAM.
The user must notice that the initial content of SRAM is unknown.

2. Index addressing mode:

In this mode, the IXAN2, IXAN1 and IXAN0 use 2-bit, 4-bit and 4-bit registers, respectively. A 10-bit register pair IXA = (IXAN2, IXAN1, IXAN0) constructs the indirect SRAM addressing pointer, and can access the whole SRAM address (00H ~ 3FFH) without using MAH. In addition, the IXA register also provide the auto-post-increment function. These registers are shown in below:

SFRs for Interrupt control：

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
IXA	**15H**	**R/W**	**xxxx**	**IXA_3**	**IXA_2**	**IXA_1**	**IXA_0**	**Read SRAM**
IXA++	**16H**	**R/W**	**xxxx**	**IXA++3**	**IXA++2**	**IXA++1**	**IXA++0**	**Read SRAM and ADR = ADR + 1**

EXIO reg. for Interrupt control：

IXAN0	16H	R/W	xxxx	IXA3	IXA2	IXA1	IXA0	IXA.N0 is SRAM Index Register
IXAN1	17H	R/W	xxxx	IXA7	IXA6	IXA5	IXA4	IXA.N1 is SRAM Index Register
IXAN2	1FH	R/W	xxxx	0	0	IXA9	IXA8	IXA.N2 is SRAM Index Register

Table 15. Registers of IXA

The SRAM related instructions are described as follows:

(2a) LD IXANx, A : The 1st(low, x=0), 2nd and 3rd(high, x=2) nibble of IXA register can be individually specified by A register, where x = 0 ~ 2 indicates the nibble position of IXA register.

(2b) LD A, IXANx : This instruction can individually read the 1st, 2nd and 3rd nibble of IXA register to A register.

(2c) "**ALU_OP** A, (IXA)", "**OP** (IXA)", "SET #D,(IXA)", "CLR #D,(IXA)", "LD A, (IXA)" or "LD (IXA), A": When access this IXA register, SRAM that pointed by IXA will be read or written. The **ALU_OP** represents ADC, SBC, CMP, OR, AND and XOR instructions. The **OP** represents INC, DEC and ADR instructions.

52#6 "**ALU_OP** A, (IXA++)", "**OP** (IXA++)", "SET #D,(IXA++)", "CLR #D,(IXA++)", "LD A, (IXA++)" or "LD (IXA++), A": When access this IXA++ register, SRAM that pointed by IXA will be read or written, and IXA will be incremented by one (auto-post-increment). The **ALU_OP** and **OP** are described as above.

For example, assume that C = 1, SRAM (115H) = 2H and SRAM (116H) = 7H.

```
LD A, #1
LD IXAN2, A
LD IXAN1, A
LD A, #5
LD IXAN0, A    ; IXA = 115H
LD A, (IXA)    ; A = SRAM (115H) = 2H
ADC A, (IXA++) ; A = A + SRAM(115H) + C = 5H, and then IXA = IXA + 1 = 116H
INC (IXA++)    ; SRAM (116H) = SRAM (116H) + 1 = 8H, IXA = IXA + 1 = 117H
```

5.3.1 Special Function Registers

The special function register consists of common I/O and extended I/O special register.

A special function register supports LD/ADC/SBC/OR/AND/XOR/INC/DEC/RLC/RRC/CMP/ADR operation. Bit set/clear can only be operated on the address range from 00H to 0FH, except indirect operation is used. The following table describes all of the common I/O SFRs.

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
STATUS	00H	R/W	00xx	INTRET	PWFG	CF	ZF	Status Register DMA_INTRET: DMA data from Interrupt backup. CF: Carry ZF: Zero PWFG: PWM interrupt flag
	01H							Reserved
IOC_PA	02H	R/W	0000	IOCA3	IOCA2	IOCA1	IOCA0	"1" = output, "0" = input of related PA bit
DATA_PA	03H	R/W	xxxx	DPA3	DPA2	DPA1	DPA0	Read data from input port and write data to output port
	04H							Reserved
	05H							Reserved
	06H							Reserved
USER1	07H	R/W	xxxx	USER1.3	USER1.2	USER1.1	USER1.0	General purpose user register
AUD_DLL	08H	W	xxxx	AUD_D.3	AUD_D.2	AUD_D.1	AUD_D.0	AUD_DLL: the bit3~0 of 12 bits PWM.
PWM_CTRL	09H	R/W	X0xx		DSEN	ENINT	ENPWM	ENPWM: 1: enable PWM 0: disable PWM ENINT: 1: enable Audio interrupt. 0 :disable Audio interrupt. DSEN: (W) : Deep Sleep Enable. 1: Enable. Store IO function. 0: Disable. (R) : Deep Sleep Flag 1: Wake up from Deep Sleep. 0: Wake up from halt or power on.
AUD_DL	0AH	W	xxxx	AUD_D.7	AUD_D.6	AUD_D.5	AUD_D.4	AUD_DL: the bit7~4 of 12 bits PWM
AUD_DH	0BH	W	xxxx	AUD_D.11	AUD_D.10	AUD_D.9	AUD_D.8	AUD_DH: the bit11~8 of 12 bits PWM
USER2	0FH	R/W	xxxx	USER2.3	USER2.2	USER2.1	USER2.0	General purpose user register
IADJ	10H	R/W	0000	-	-	ADJ1	ADJ0	ADJ[1:0]: Adjust the frequency, when Enable option "OPTADJ". ADJ[1:0] / Frequency % 00 / -4% 01 / -2% 10 / 2% 11 / 4%
BZPWEN	11H	R/W	0000	BZPWEN1	-	S2S	-	BZPWEN1:PA0 buzzer wakeup and PWMN speaker wakeup control. 1: enable buzzer and speaker wakeup 0: disable buzzer and speaker wakeup S2S: PWM input Data format 1: Inverting Audio Data without sign bit. AUD_D.11 is sign bit. 0: Normal Audio Data format

<table>
<tr><td>PWMWK</td><td>12H</td><td>R/W</td><td>0000</td><td>PWMWKFG</td><td>CLAPSEL1</td><td>CLAPSEL0</td><td>PWMWKEN</td><td>PWMWKFG:PWMN speaker wakeup flag
1: PWMN speaker wakeup occurred
0: no PWMN speaker wakeup

CLAPSEL[1:0]: Sensitivity of Speaker wakeup or Buzzer wakeup.
<table><tr><td>CLAPSEL[1:0]</td><td>Sensitivity</td></tr><tr><td>00</td><td>level1(high sen.)</td></tr><tr><td>01</td><td>level 2</td></tr><tr><td>10</td><td>level 3</td></tr><tr><td>11</td><td>level 4</td></tr></table>
PWMWKEN: PWMN speaker wakeup control
1: enable
0: disable</td></tr>
<tr><td>PWMWKS</td><td>13H</td><td>R/W</td><td>0001</td><td>PWMWKS
(read only)</td><td>BZPWEN2</td><td>DETEN</td><td>GINTEN</td><td>PWMWKS:
Read external PWMN pin status to confirm PWMN wakeup signal.

BZPWEN2: PA0 buzzer wakeup and PWMN speaker wakeup control
1: enable buzzer and speaker wakeup
0: disable buzzer and speaker wakeup

DETEN: PA0 pin special input level detector control
1: Enable.
0: Disable.

GINTEN :
1: Global interrupt enable.
0: Global interrupt disable.</td></tr>
<tr><td>IXA</td><td>15H</td><td>R/W</td><td>xxxx</td><td>IXA_3</td><td>IXA_2</td><td>IXA_1</td><td>IXA_0</td><td>Read SRAM</td></tr>
<tr><td>IXA++</td><td>16H</td><td>R/W</td><td>xxxx</td><td>IXA++3</td><td>IXA++2</td><td>IXA++1</td><td>IXA++0</td><td>Read SRAM and ADR = ADR + 1</td></tr>
<tr><td>WATCH</td><td>17H</td><td>W</td><td>xxxx</td><td></td><td></td><td></td><td></td><td>Writing this register means reset watch dog timer if this timer is enabled by option.</td></tr>
<tr><td>DMA0</td><td>18H</td><td>R/W</td><td>xxxx</td><td>DMA0.3</td><td>DMA0.2</td><td>DMA0.1</td><td>DMA0.0</td><td rowspan="4">DMA0~4, five register built a 19 bit addressing space for read DROM 8 bits data, DMA0 is lowest nibble, DMA4 is highest nibble of DROM address</td></tr>
<tr><td>DMA1</td><td>19H</td><td>R/W</td><td>xxxx</td><td>DMA1.3</td><td>DMA1.2</td><td>DMA1.1</td><td>DMA1.0</td></tr>
<tr><td>DMA2</td><td>1AH</td><td>R/W</td><td>xxxx</td><td>DMA2.3</td><td>DMA2.2</td><td>DMA2.1</td><td>DMA2.0</td></tr>
<tr><td>DMA3</td><td>1BH</td><td>R/W</td><td>xxxx</td><td>DMA3.3</td><td>DMA3.2</td><td>DMA3.1</td><td>DMA3.0</td></tr>
<tr><td>DMDL</td><td>1CH</td><td>R/W</td><td>xxxx</td><td>DMDL.3</td><td>DMDL.2</td><td>DMDL.1</td><td>DMDL.0</td><td>Low nibble of DROM data read from this address.
Store low nibble DROM Data for write to Flash.</td></tr>
<tr><td>DMDH</td><td>1DH</td><td>R/W</td><td>xxxx</td><td>DMDH.3</td><td>DMDH.2</td><td>DMDH.1</td><td>DMDH.0</td><td>High nibble of DROM data read from this address.
Store high nibble DROM Data for write to Flash.</td></tr>
</table>

DMA4 1EH

U: unchanged X: unknown value R/W: readable & writeable R: readable only W: writeable only

<table>
<tr><th>Symbol</th><th>Addr</th><th>R/W</th><th>Reset</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th><th>Description</th></tr>
<tr><td>MAH</td><td>00H</td><td>R</td><td>0000</td><td>MAH.3</td><td>MAH.2</td><td>MAH.1</td><td>MAH.0</td><td>SRAM MAH</td></tr>
<tr><td>WKRTC</td><td>01H</td><td>R/W</td><td>0000</td><td>RTCFG</td><td>INTVEB</td><td>RTCS1</td><td>RTCS0</td><td>RTCFG: HRTC overflow flag
0: HRTC overflow not occurred.
1: HRTC overflow occurred, it could be cleared by software.

INTVEB:
0: The Interrupt Vector Keeps last set value, default value 0x008
1: Interrupt Vector will load from PMA3~ PMA0 registers

RTCS1, RTCS0: HRTC wake up period selection.
<table>
<tr><th>RTCS1</th><th>RTCS0</th><th>WKRTC period (± 50%)</th></tr>
<tr><td>0</td><td>0</td><td>16ms</td></tr>
<tr><td>0</td><td>1</td><td>32ms</td></tr>
<tr><td>1</td><td>0</td><td>64ms</td></tr>
<tr><td>1</td><td>1</td><td>128ms</td></tr>
</table>
</td></tr>
<tr><td></td><td>02H~ 03H</td><td>R/W</td><td>0000</td><td></td><td></td><td></td><td></td><td>Reserved</td></tr>
<tr><td>LVD_CTL2</td><td>04H</td><td>R/W</td><td>0000</td><td>LVDEN</td><td>LVDFG (Read only)</td><td>D2ENB</td><td>LVDS4</td><td>LVDEN: LVD enable control
1: enabled
0: disabled
LVDFG: Low voltage detector flag
0: no low voltage occurred</td></tr>
</table>

	0DH	R/W	0000					Reserved
MUL_AL	0EH	R/W	xxxx	MAI3 P3	MAI2 P2	MAI1 P1	MAI0 P0	12 x8 bits Multiplication registers Multiplier is 12 bits MAI11~MAI0 Multiplicand is 8 bits MBI7~MBI0 Multiplication product is 20 bits P19~P0 W : multiplier low nibble MAI3~MAI0 R: multiplication product P3~P0
MUL_AM	0FH	R/W	xxxx	MAI7 P7	MAI6 P6	MAI5 P5	MAI4 P4	W : multiplier middle nibble R : multiplication product P7~P4
MUL_BL	11H	R/W	xxxx	MBI3 P15	MBI2 P14	MBI1 P13	MBI0 P12	W :12 bits multiplicand low nibble R : multiplication product P15~P12
MUL_BH	12H	R/W	xxxx	MBI7 P19	MBI6 P18	MBI5 P17	MBI4 P16	W :12 bits multiplicand middle nibble R : multiplication product P19~P16
SATV	13H	R/W	xxxx	SATV.3	SATV.2	SATV.1	SATV.0	SATV register is used to shape PWM data, input format is 16 bit and output is 12 bit. Writing SATV register four times configure 16-bit SATV input data, and reading SATV 12 bit data by three times.
	14H							
	15H							
IXAN0	16H	R/W	xxxx	IXA3	IXA2	IXA1	IXA0	IXA.N0 is SRAM Index Register
IXAN1	17H	R/W	xxxx	IXA7	IXA6	IXA5	IXA4	IXA.N1 is SRAM Index Register
PMA0	18H	R/W	xxxx	PMA0.3	PMA0.2	PMA0.1	PMA0.0	PMA0~3 four registers built a 14 bit address for read PROM 12 bits data, PMA0 is lowest nibble, PMA3 is highest nibble of PROM address
PMA1	19H	R/W	xxxx	PMA1.3	PMA1.2	PMA1.1	PMA1.0	
PMA2	1AH	R/W	xxxx	PMA2.3	PMA2.2	PMA2.1	PMA2.0	
PMA3	1BH	R/W	000x	0	0	PMA3.1	PMA3.0	
PMDL	1CH	R	xxxx	PMDL.3	PMDL.2	PMDL.1	PMDL.0	Low nibble data output when read PROM
PMDM	1DH	R	xxxx	PMDM.3	PMDM.2	PMDM.1	PMDM.0	High nibble data output when read PROM
PMDH	1EH	R	xxxx	PMDH.3	PMDH.2	PMDH.1	PMDH.0	High nibble data output when read PROM
IXAN2	1FH	R/W	xxxx	0	0	IXA9	IXA8	IXA.N2 is SRAM Index Register

Table 17 Extended I/O Special Function Registers

5.4 Interrupt Vector Address

Vector	Address
RESET	00H
OPTION	02H
WAKEUP	04H
TEST	06H
INT	Interrupt Vector is 0x008 after power on. If INTVEB register is set to "1" then Interrupt Vector will load from PMA3~PMA0 registers.

Table 18. Interrupt Vectors

5.5 Interrupt Controller

There is only one interrupt entry point in this chip. Normally interrupt period is 32000Hz. It can be changed to 64000Hz by option "PWM64K". Program will jump to address $008h when an interrupt occurs.

SFRs for Interrupt control：

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
STATUS	00H	R/W	00xx	INTRET	**PWFG**	CF	ZF	PWFG: PWM interrupt flag
PWM_CTRL	09H	R/W	xxxx			**ENINT**	**ENPWM**	ENPWM: 1: enable PWM 0: disable PWM ENINT: 1: enable Audio interrupt. 0 :disable Audio interrupt.

PWMWKS	13H	R/W	0001	PWMWKS	BZPWEN2	DETEN	**GINTEN**	GINTEN : 1: Global interrupt enable. 0: Global interrupt disable.

EXIO reg. for Interrupt control：

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
WKRTC	01H	R/W	0000	RTCFG	**INTVEB**	RTCS1	RTCS0	INTVEB: 0: Interrupt Vector keeps last value from PMA, or default value is 0x008 if INTVEB never set high after power up. 1: Interrupt Vector will load from PMA3~ PMA0 registers
	14H	R/W						

Table 19: Interrupt control registers

There are two interrupt control register GINTEN and INTEN must be set high before using PWM interrupt. If an interrupt request (PWFG=1) occurred, then interrupt will be accepted on next clock. On that instant, current (next) program counter, PCDH, MAH and C/Z will be stored in special hardware registers, and program counter will be loaded with entry address of that interrupt. All these are done in one clock. Interrupt enable bit will be cleared too. As long as program enters interrupt service, interrupt enable bit is cleared. It's no need to clear interrupt enable flag in interrupt routine. But hardware will not clear interrupt request flag (PWFG). Software is required to clear it.

When interrupt service routine is done, an RETI shall be executed. This instruction will restore stored program counter, PCDH, MAH and C/Z, will set interrupt enable bit=1 also. (Note that RETI is different from RETS.) Interrupt request can be accepted only when enable bit be set. Note that only one level of INT routine can be used.

The interrupt vector can be customized by programmers. To enable this function, programmer should place the new vector to PAM3~PAM0 registers and set INTVEB to high then cleared to low before enabling interrupt.
Customized interrupt vector operating procedure as shown below:

1. Assign new interrupt vector to PMA3~PMA0 registers.
2. Set INTVEB to high means to store PMAx registers to Interrupt Vector register.
3. Clear INTVEB to low means to release PMAx registers and PMAx registers are available for another function.
4. To enable interrupt

Customized interrupt vector example program

```
LD      A, #INV_EXP.N3
LD      EXIO(PMA3), A
LD      A, #INV_EXP.N2
LD      EXIO(PMA2), A
LD      A, #INV_EXP.N1
LD      EXIO(PMA1), A
LD      A, #INV_EXP.N0
LD      EXIO(PMA0), A           // Set PMA = 0x0123

LD      A, EXIO(WKRTC)
OR      A, #0100B
LD      EXIO(WKRTC), A          // Setting Interrupt Vector to address 0x0123
AND     A, #1011B
LD      EXIO(WKRTC), A          // Keep last value from PMA, Interrupt Vector is address 0x0123

ORG     0X0123
```

INV_EXP:
…

5.6 Operation Mode

There are two operation modes in this chip. The state diagram of three MCU operation modes is shown below.

1. NORMAL Mode: In normal mode, system clock oscillator is running, and MCU clock source is come from main oscillator. In NORMAL mode, MCU will go to halt mode after HALT instruction executed.

2. HALT Mode Mode: In HALT mode, the MCU clock stops, users can't change the operation mode when in halt mode. It will go back to NORMAL mode (Program counter=004) when I/O wakeup or reset occurred. Please refer to the section of" Halt Mode & Wake up" for the detailed HALT mode description.

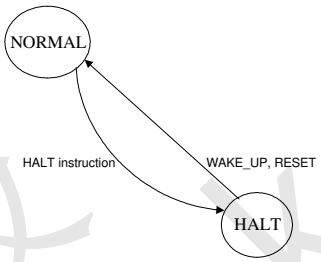


Figure 4. Operation Mode

5.6.1 Deep Sleep Mode

In Deep Sleep Mode, the IC will cut off power except I/O wakeup function, so user must backup data to DROM by user code before into Deep Sleep mode. Users can't change the operation mode when in Deep Sleep mode.

When "DEEP_SIG" be set:
1. I/O status cannot be changed.
2. I/O step will remain ready to enter Deep Sleep mode.

When "DEEP_SIG" be clear:
1. I/O status can be changed after 1ms when clear DEEP_SIG. User can polling DEEP_SIG to check it.

PWM_CTRL	09H	R/W	X0xx		DEEP_SIG	ENINT	ENPWM	ENPWM: 1: enable PWM 0: disable PWM ENINT: 1: enable Audio interrupt. 0: disable Audio interrupt. DEEP_SIG: DEEP Signal 0: Wakeup from Halt mode or power on. 1: Wakeup from Deep sleep.

5.7 Halt Mode & Wake up

After the system enters the HALT mode it can be woken up from one of various source listed as below:
(1) An external I/O pin rising edge.
(2) A HRTC overflow
(3) A RC wakeup
(4) A buzzer wakeup
(5) A speaker wakeup

The MCU will go into HALT mode (program counter and system clock stop) when HALT instruction executed. It provides a power saving mode for those applications requiring a very low stand-by current.

5.7.1 I/O wake up

Each pin on PA[3:0] or PE[3:0] can be setup using the correspond option(WAKEUPBA3~0, WAKEUPBE3~0) to permit positive transition on the pin to wake up the system. The program counter will be 08H when HALT instruction executed immediately, then program counter will go to next address after 64 stable clock(system clock) when wake up condition occurred. Reset signal will release HALT state and execute reset procedure

because reset is first priority when in HALT mode, so program counter will from 04h to 00h, program counter goes to next address after 64 stable clock cycles. Furthermore, the SRAM will keep their previous data without changing in this mode.

5.7.2 HRTC(HALT mode RTC) wake up

HALT mode RTC (HRTC) works on HALT mode only if RTC_EN option is enabled. The HRTC period has four kinds of period shown as below. RTCFG register set high indicates HRTC wake up occurred.

WKRTC	01H	R/W	0000	RTCFG	INTVEB	RTCS1	RTCS0	RTCS1, RTCS0: HRTC wake up period selection. RTCFG: WKRTC overflow flag 0: HRTC overflow not occurred. 1:HRTC overflow occurred, it could be cleared by software.

Table 20. HRTC control register

HTRC wake up period selection.

RTCS1	RTCS0	HRTC Period (± 50%)
0	0	16ms
0	1	32ms
1	0	64ms
1	1	128ms

Table 21. HTRC wake up period selection

5.7.3 Special wakeup (PA0 RC, PA0 buzzer, PWMN speaker)

There are three wakeup called special wakeup. They are PA0 RC wakeup, PA0 buzzer wakeup and PWMN speaker wakeup.

Common I/O register

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
BZPWEN	11H	R/W	0000	**BZPWEN1**	-	-		BZPWEN1:PA0 buzzer wakeup and PWMN speaker wakeup control 1: enable buzzer and speaker wakeup 0: disable buzzer and speaker wakeup
PWMWK	12H	R/W	0000	**PWMWKFG**	**CLAPSEL1**	**CLAPSEL0**	**PWMWKEN**	PWMWKFG:PWMN speaker wakeup flag 1: PWMN speaker wakeup occurred 0: no PWMN speaker wakeup CLAPSEL[1:0]: Sensitivity of Speaker wakeup or Buzzer wakeup. CLAPSEL[1:0] / Sensitivity 00 / level1(high sen.) 01 / level 2 10 / level 3 11 / level 4 PWMWKEN: PWMN speaker wakeup control 1: enable 0: disable
PWMWKS	13H	R/W	0001	**PWMWKS (read only)**	**BZPWEN2**	DETEN	GINTEN	PWMWKS: Read external PWMN pin status to confirm PWMN wakeup signal. BZPWEN2: PA0 buzzer wakeup and PWMN speaker wakeup control 1: enable buzzer and speaker wakeup 0: disable buzzer and speaker wakeup

Table 22 Special wakeup control registers

PA0 RC wakeup

It provides a PA0 RC wakeup function.PA0 supports an analog input buffer with schmitt input circuit, it is enabled by RCWK option and supports low power consumption in halt mode, if PA0 analog signal keep about 0.5VCC, so this function is suitable for recycle wakeup MCU by external RC time constant, external RC time constant is easy built by VCC connected to R and serial with C to VSS.
The block diagram and electrical characteristic of PA0 analog input buffer shown below. It takes about 5uA when PA0 input signal is 2.5v @VCC=5V in halt mode, otherwise, it takes about 500uA current if RCWK option is disabled.
This block function is enabled when RCWK option enabled and set IOC_PA=0000B by hardware after power on reset.

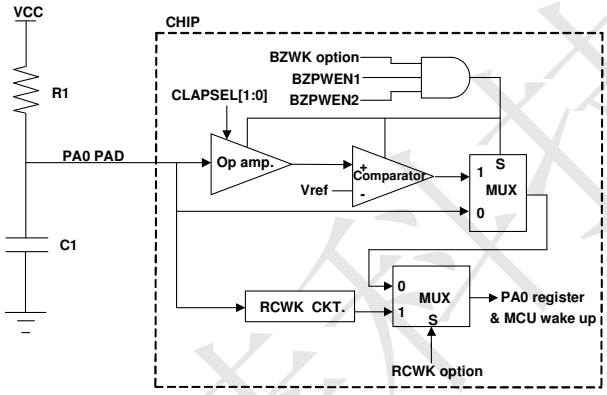


Figure 5. PA0 RC wakeup block diagram

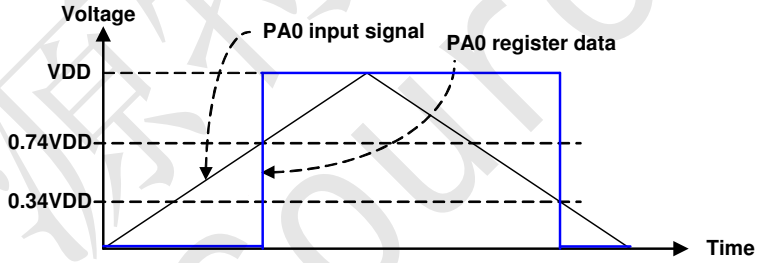


Figure 6. PA0 input signal and PA0 register data

PA0 buzzer wakeup

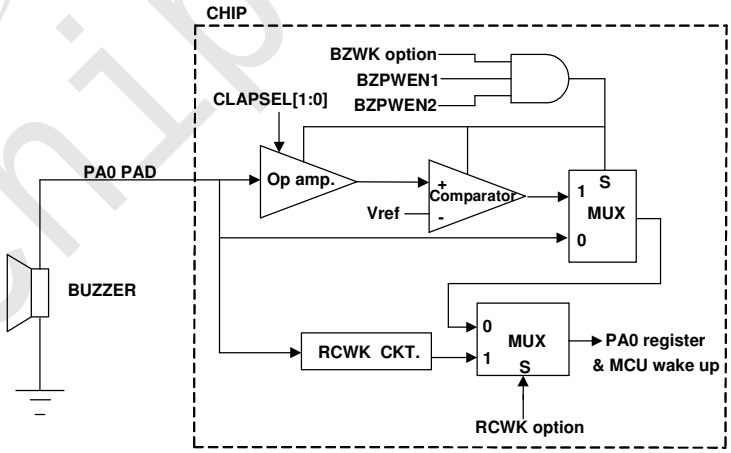


Figure 7. PA0 buzzer wake up block diagram

MCU can be wakeup by a small signal on PA0 PIN which connected a buzzer to VSS. It built-in an OP amplifier for wakeup trigger, the pull down 1M and 50k resistor will be disabled by hardware circuit when "BZWK" option is enabled automatically. To enable this function, please set BZPWEN1 and BZPWEN2 to "1" and set BZWK option enabled on IDE tool.

PA0 Buzzer wakeup control register and Option

BZWK Option	BZPWEN1	BZPWEN2	PA0 buzzer wakeup function
0	0	0	Disabled
1	1	1	Enabled
Others			Disabled

Table 23. PA0 Buzzer wakeup control

There are 4 sensitivity level for speaker wakeup and buzzer wakeup control.
Sensitivity selection

CLAPSEL[1:0]	Sensitivity
00	level1(high sen.)
01	level 2
10	level 3
11	level 4

Table 24 Sensitivity selection

Buzzer wakeup example program

```
LD A, #0000B
LD (PWMWK), A        ; assign sensitivity level CLAPSEL[1:0] = 00
LD A, #1000B
LD (BZPWEN), A       ; assign BZPWEN1 =1
LD A, #0101B
LD (PWMWKS), A       ; assign BZPWEN2 = 1
…
…

HALT
```

PWMN speaker wakeup

MCU can be wakeup by a small signal on PWMN PIN which connected a speaker to PWMP PIN. The PWMWKFG register indicate PWMN speaker wakeup occurred or not. In order to confirm PWMN wakeup signal, we can read external PWMN pin status from PWMWKS register. To enable this function, please set BZPWEN1, BZPWEN2 and PWMWKEN to high.

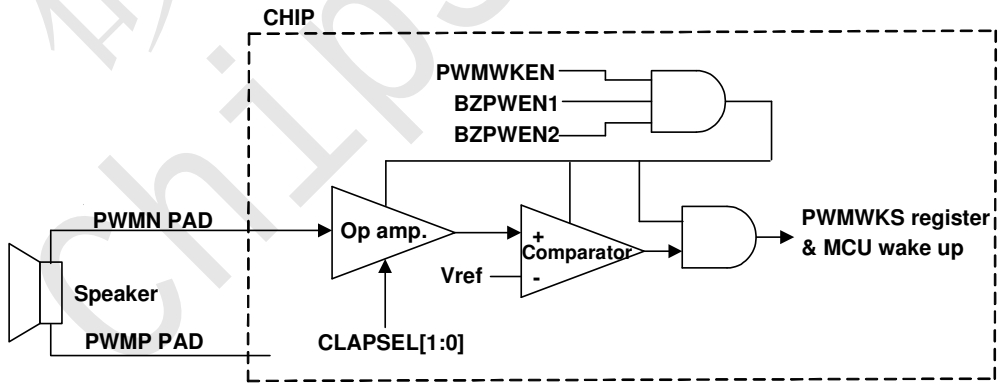


Figure 8. Speaker wake up block diagram

PWM wakeup control register

PWMWKEN	BZPWEN1	BZPWEN2	Speaker wake up function
0	0	0	Disabled
1	1	1	Enabled
Others			Disabled

Table 25. PWM wakeup control

PWMN speaker wakeup example program

```
LD A, #0001B
LD (PWMWK), A        ; assign PWMWKEN=1 and sensitivity level CLAPSEL[1:0] = 00
LD A, #1000B
LD (BZPWEN), A       ; assign BZPWEN1 =1
LD A, #0101B
LD (PWMWKS), A       ; assign BZPWEN2 = 1
…
…

HALT
```

5.8 Watch Dog Timer Reset (WDT)

The watch dog timer (WDT) is used to reset whole chip when detect unexpected execution sequence caused by accident condition, avoiding dead lock of MCU program. WDT will be reset when wake-up from halt, after reset or software clears it.

The watch dog timer is a simple counter. WDT period is about 0.13 sec. (Clock source by LRCOSC = 32.000K Hz ± 50%) Software must run a "clear watch dog timer" (write to WDT) instruction before this timer time out when WDT enabled. It will generate a reset signal to reset whole system when WDT over flow. Assembler will check user program about the "reset watch dog" instructions. If more than one "reset watch dog" is found in program, assembler will generate a warning (program code will still be generated). "HALT" instruction will reset watch dog timer. The reset watch dog timer sequence is strongly recommended as:

```
WATCHDOG:    LD      (17H), A
             LDPCH   WATCHDOG
             LDMAH#n
```

5.9 8/10/12 bits PWM

One is 8 bit output, the sec. is 10 bit output, and the other is 12 bit output. The highest of input data is signed bit: ‘0’ represents positive, ‘1’ represents negative. The 12 bits PWM data (AUD_D.11~ AUD_D.0) consist of AUD_DH, AUD_DL, AUD_DLL three registers shown above. The AUD_D.11~ AUD_D.0 registers must be assigned before PWM is enabled.

It provides SATV and DSI block to improve sound and speech effect without software preparation. The SATV block solves signal saturation when multi-channel melody or speech mixing together and make sound more smoothly. It can accept 16 bit signal input and generate 12bits signal output for next block DSI. The DSI block means Delta-Sigma Interpolation makes PWM play up to 256 KHz by interpolation skill.

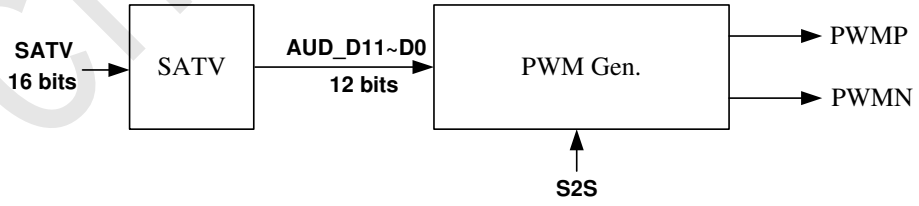


Figure 9. PWM block diagram

Writing SATV register four times configure 16-bit SATV input data, and reading SATV 12 bit data by reading SATV register three times.

Common I/O

Symbol	Addr	R/W

The system reset comes from three signals which are power on reset, low voltage reset(LVR), external RESETB pin and WDT overflow reset.

For normal operation of this chip, a good reset is needed. The operation frequency of MCU will go back to NORMAL mode when reset occurred in HALT mode.

5.11 System Clock Oscillator

This chip MCU is typically operated on 13.1072MHz which is generated from internal RC oscillator 65.536MHz.

5.12 I/O Ports

There is one I/O port, PA0~PA3, whose input/output direction are defined by IOC_PA. The wake-up functions of PA0~PA3 are enabled or disabled by option. Their 1M/50k pull down resistors are optional.
In order to achieve touch function, CST4A0XXA/B series support 220k pull down resistors. These resistors can be enabled by using 50k pull down resistor registers after “PD220K” is enabled. The 220K resistance value is almost fixed value when VCC change from 2.4V to 5.5V.
All I/O ports provides rising or falling edge wake up function. This double edge wake up function can be enabled by “BIWK” option.

5.12.1 Port PA (input/output)

The Port A are 4-bit bidirectional I/O ports. Their directions can be defined by IOC_PA bit by bit.
The following table describe the SFRs associated with Port A.

Symbol	Addr	R/W	Reset	D3	D2	D1	D0	Description
IOC_PA	02H	R/W	0000	IOCA3	IOCA2	IOCA1	IOCA0	“1” = output, “0” = input of related PA bit
DATA_PA	03H	R/W	xxxx	DPA3	DPA2	DPA1	DPA0	Read from input port and write to output port

Table 27. SFRs of Ports PA

In output mode, the data can be written out to external pin, and reading this output port will read the data internal latch. Pull-down resistor will be disabled when output mode is selected.
In input mode, Port A data can be read from external pin, and they are attached with 1M/50K pull-down resistor or not according to the options.
In addition, each pin of Port A also can be accompanied with wakeup function according to the options. In HALT mode, if some bits of Port A are accompanied with wakeup function, any rising edge occurred on that pin will wake up system and turn on oscillator, and the program counter of MCU will jump to the address 04H. This device will start to execute the wake-up sub-routing.

PA3 is provided with 38KHz modulator
I/O port PA3 built-in a 38KHz modulator combined with register DATA_PA bit 3, this function is enabled by F38K option. If F38K option is enabled, I/O port PA3 will output 38KHz clock signal when PA3 is configured as output port and DATA_PA bit3 = 1. PA3 output low when PA3 is configured as output port and DATA_PA bit3 = 0.
The PA3 output pad will be forced to low state automatically by hardware control when in halt mode for avoid external IR LED destroyed. Besides, the F38K option disabled, PA3 is a normal I/O port.

PA0 provides two wakeup function, one is RC wakeup and the other is Buzzer wakeup. Please see 5.9.2 section for detail description.

5.13.1 Indirect Jump Instruction (JMPX)

JMPX instruction provides a Indirect jump to the physical address. The physical address 13 bits consists of two parts, the MSB comes from LDPCH’s MSB and the lower 12bits specified from the content of X indirect

pointer. X indirect pointer combines with PMA3~PAM0 registers, PMA3 is high nibble and PMA0 is low nibble.

The JMPX instruction procedure as shown below:

1. Select a X indirect pointer and assign PMA3~ PMA0 registers.
2. Decide a physical address 13 bit (A12~A0) you want to jump.
3. Assign MSB(A12) of physical address by load LDPCH to accumulator.
4. Assign physical address (A11~A0) 12 bits to PROM by PDW instruction, and this address is pointed by PMA3~PMA0.
5. Jump to the physical address by JMPX instruction.

The following example shows how to use JMPX instruction.
Assume program needs jump to physical address 0700h by JMPX instruction. X indirect pointer is assigned in $0F80h. So the content of $0F80h is 700h. Due to physical address MSB is “0”, so the label LDPCH to load must be located at the lower 4 K address in PROM.

```
    LD A,#0h                ; assign X indirect pointer address
    LD (PMA0),A
    LD A,#8h
    LD (PMA1),A
    LD A,#Fh
    LD (PMA2),A
    LD A,#0h
    LD (PMA3),A
    LDPCH    MSB_0          ; Assign MSB of physical address to “0”, MSB(A12)=0
    JMPX                    ; Jump to address $700h

    ORG 0700H
    LDPCH    MSB_1          ; Assign MSB of physical address to “1”, MSB(A12)=1
    JMPX                    ; Jump to address $1700h

    ORG 1700H
    LD  A, #0H
    …
    …

    ORG 0123H
MSB_0:                      ; the lower 4 K address in PROM  for MSB(A12) set to “1”
….
    ORG 1123H
MSB_1:                      ; the upper 4 K address in PROM for MSB(A12) set to “0”
…..
    ORG 0F80H               ; X indirect pointer address
PDW   700H                  ; Assign lower physical address 12 bits (A11~A0)
```

5.13.2 Direct Jump Instruction (JMPPMA)

JMPPMA instruction provides a direct jump to the physical address. The physical address 13 bits consists of PMA3~PAM0 registers.

The JMPPMA instruction procedure as shown below:

1. Decide a absolute address you need to jump and assign PMA3~ PMA0 registers .
2. Jump to the absolute address by JMPPMA instruction.

The following example shows how to use JMPPMA instruction.

```
LD A, #0h              ; assign jump address
 LD (PMA0),A
 LD A, #8h
 LD (PMA1),A
 LD A, #Fh
 LD (PMA2),A
 LD A, #0h
 LD (PMA3),A

 JMPPMA                ; jump to address $0F80h


 ORG 0F80H
 LD  A, #0H
 …
 …
```

5.13.3 Indirect call Instruction (CALLX)

CALLX instruction provides a Indirect call to the physical address. The physical address 13 bits consists of two parts, the MSB comes from LDPCH's MSB and the lower 12bits specified from the content of X indirect pointer. X indirect pointer combines with PMA3~PAM0 registers, PMA3 is high nibble and PMA0 is low nibble.

The CALLX instruction procedure as shown below:

1. Select a X indirect pointer and assign PMA3 ~ PMA0 registers .
2. Decide a physical address 13 bit (A12~A0) you want to call.
3. Assign MSB (A12) of physical address by load LDPCH to accumulator.
4. Assign physical address (A11~A0) 12 bits to PROM by PDW instruction, and this address is pointed by PMA3~PMA0.
5. CALL to the physical address by CALLX instruction.

The following example shows how to use CALLX instruction.
Assume program needs call to physical address 0600h by CALLX instruction. X indirect pointer is assigned in $0ABCh. So the content of $0ABCh is 600h. Due to physical address MSB is "0", so the label LDPCH to load must be located at the lower 4 K address in PROM.

```
 LD A,#Ch                ; assign X indirect pointer address
 LD (PMA0),A
 LD A,#Bh
 LD (PMA1),A
 LD A,#Ah
 LD (PMA2),A
 LD A,#0h
 LD (PMA3),A
 LDPCH    MSB_0          ; Assign MSB of physical address to "0", MSB(A12)=0
 CALLX                   ; Call to address $600h
```

```
    LDPCH      MSB_1          ; Assign MSB of physical address to “1”, MSB(A12)=1
    CALLX                     ; Call to address $1600h

    ORG 0600H
    …
    …
    RETS                      ;  CALL RETRUN

    ORG 1600H
    …
    …
    RETS                      ;  CALL RETRUN

    ORG 0123H
MSB_0:                        ; the lower 4 K address in PROM  for MSB(A12) set to “1”
….
    ORG 1123H
MSB_1:                        ; the upper 4 K address in PROM for MSB(A12) set to “0”
…..
    ORG 0ABCH                 ; X indirect pointer address
PDW    600H                   ; Assign lower physical address 12 bits (A11~A0) = 600h
```

5.13.4 Direct call Instruction (CALLPMA)

CALLPMA instruction provides a direct CALL to the p

```
    ...
    RETS            ; CALL RETRUN

    ORG 1CDEH
TABLE_Y1:
    ...
    RETS            ; CALL RETRUN
```

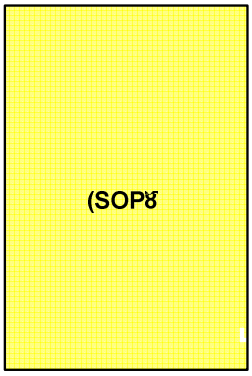
5.14 12Bit X 8Bit Multiplier (signed x unsigned)

An 12 bit x 8 bit (signed x unsigned) multiplier is provided for data processing. This multiplier contains 5 nibble registers, two nibble **MUL_AL** and **MUL_AH** are for input A register, three nibbles **MUL_BL**, **MUL_BM and MUL_BH** are for input B register. The product is a 20 bits data stored in P19 ~ P0. This multiplication starts as long as high nibble(MUL_BH) of B is stored. Note that the content of A and B registers will be destroyed during multiplication procedure.

Symbol	Addr.	R/W	Reset	D3	D2	D1	D0	Description
MUL_AL	0EH	R/W	xxxx	MAI3 P3	MAI2 P2	MAI1 P1	MAI0 P0	12 x8 bits Multiplication registers Multiplier is 12 bits MAI11~MAI0 Multiplicand is 8 bits MBI7~MBI0 Multiplication product is 20 bits P19~P0 W : multiplier low nibble MAI3~MAI0 R: multiplication product P3~P0
MUL_AM	0FH	R/W	xxxx	MAI7 P7	MAI6 P6	MAI5 P5	MAI4 P4	W : multiplier middle nibble R : multiplication product P7~P4
MUL_AH	10H	R/W	xxxx	MAI11 P11	MAI10 P10	MAI9 P9	MAI8 P8	W : multiplier high nibble R : multiplication product P11~P8 Multiplication will go start after write data to MUL_AH register automatically
MUL_BL	11H	R/W	xxxx	MBI3 P15	MBI2 P14	MBI1 P13	MBI0 P12	W :12 bits multiplicand low nibble R : multiplication product P15~P12
MUL_BH	12H	R/W	xxxx	MBI7 P19	MBI6 P18	MBI5 P17	MBI4 P16	W :12 bits multiplicand middle nibble R : multiplication product P19~P16

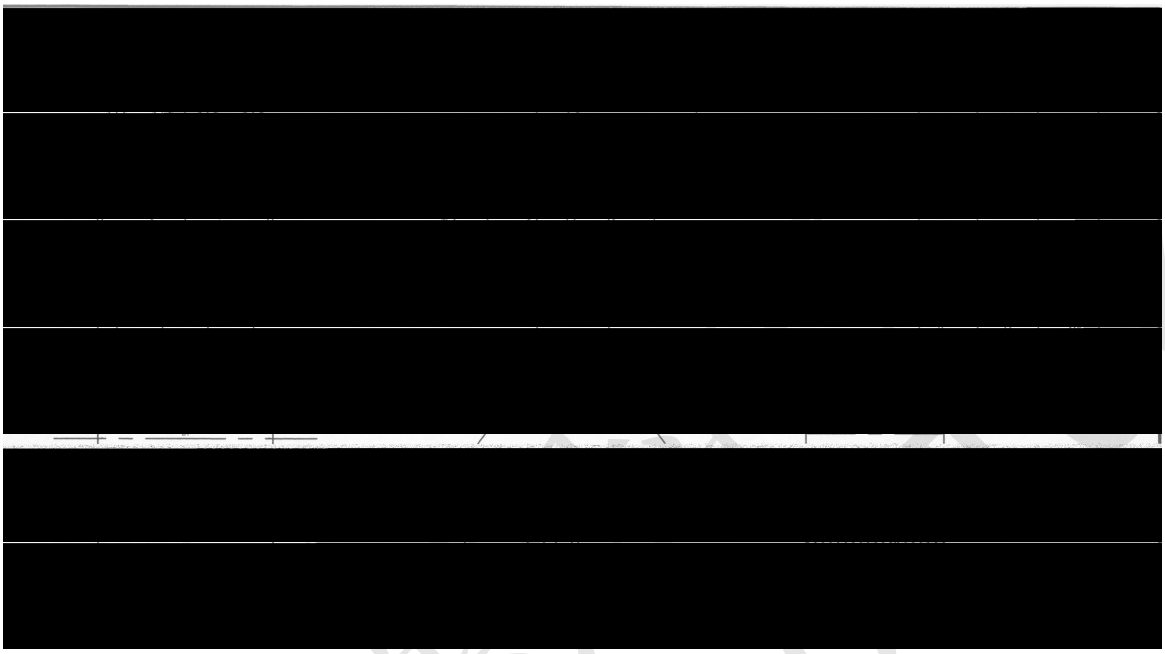
Table 28. SFR of Multiplier

6. CST4A0XXA/B The Application Circuit



7. CST4A0XXA/B Package Information

7.1 SOP8



8. CST4A0XXA/B Option Registers table

Option Name	Function Description
PROM2K	PROM size selection 2K
PROM4K	PROM size selection 4K
PROM6K	PROM size selection 6K
PROM8K	PROM size selection 8K
RCWK	PA0 RC wakeup function.
BZWK	Buzzer wakeup function.
WAKEBA	Wake-up enable for PA3~PA0 respectively.
BIWK (CST4A0XXB only)	double edge wake up function Enable = double edge Disable = rising edge
PD50KPA	50K Ohm pull down resistor for PA3~PA0 respectively.
PD1MPA	1M Ohm pull down resistor for PA3~PA0 respectively.
PD220K	Change 50K Ohm pull down resistor to 220K Ohm Enable=220K Ohm, Disable=50K Ohm.
WDGENB	Watch dog timer function.
WDHEN	Watch dog timer in halt mode function.
OTPLOCK	Security control, Disable = Insecurity, Enable = Security.
OPTADJ	OSCADJ select function
HDEN	Driving capacity of output port, Enable = 20mA sink and source capacity. Disable = 4mA sink and source capacity.
INTMASK	Interrupt mask Enable = Into Interrupt, set ENPWM and ENINT both. Disable = Into Interrupt, set ENINT only.
RTC_EN	Enable RTC in halt mode function
F38K	PA3 built-in a 38KHz modulator function
OTPREV	Fill the unused PROM with 0xFFF
PWM64K	Enable = interrupt period is 64000Hz Disable = interrupt period is 32000Hz
DEEPSLEEP	Deep Sleep function.

9. CST4A0XXA/B The Revision History

Version	Description	Page	Date
1.0	Established		2021-05-05
1.1	Edit voltage range	1	2021-05-21
	Edit Table 17 Extended I/O Special Function Registers	16	
1.2	Edit CST4A0XXA/B series selection table.	2	2021-09-22
	Rename Registers Symbol: AUD_DL → AUD_DLL.	13, 22, 23	
	Rename Registers Symbol: AUD_DM → AUD_DL.	13, 22, 23	
	Rename Registers Symbol: SYS0 → PWM_CTRL.	13,16,18,22	
	Rename Registers Symbol: WDT → WATCH.	14	
	Added Option Registers table	32	

Table 29. Revision History