



深圳市矽源特科技有限公司  
ShenZhen ChipSoureTek Technology Co.,Ltd.

---

# CST1608A(C)

**Data Sheet**  
**V1.6**

512K Embedded Flash  
Hi-Performance 16-bit Speech Processor



---

## CST1608A(C) 16-bit Speech Processor

### 1. CST1608A(C) General Description

The TxP16S2™ is a high performance 16-bit MCU, running up to 32MHz and provided with 512K FLASH and total 4K SRAM for high performance process of audio algorithm. It is the new generation computational kernel for Flash Speech series. It has initially aimed at the areas of speech application to demonstrate its profession. TxP16S2 furnish with a fast unit, which allows calculation instructions to be issued with access memory simultaneously during one cycle. The CST1608A(C) is equipped with TxP16S2 and integrating input/output ports, Audio PWM/DAC, Timer and Low Voltage Reset...etc on a chip.

Furthermore, CST1608A(C) extend its external device connection capability such as Serial ROM/Flash. The internal memory capacity includes 512Kx16 program/data FLASH plus 4Kx16 working SRAM.

矽源特科技  
ChipSourceTek



## CST1608A(C) 16-bit Speech Processor

### 2. CST1608A(C) Features

- ◆ **Notice: CST1608C only supports SOP16-3 and SSOP24-2 packages.**
- ◆ **Notice: CST1608A(C) do not support ultra-deep sleep.**
- ◆ **Notice: PortC2 are forbidden to be connected with VCC or pull-up resistance.**
- ◆ **Notice: CST1608A(C) does not support buzzer and speaker wake up, so MISC1 bit10, MISC2 bit 12 and MISC2 bit14 should be set to 0.**
- ◆ High-performance RISC TxP16S2 CPU
  - wide working frequency and voltage 4Mhz ~32Mhz@1.8Volt ~ 5.5Volt
  - Operation frequency is programmable by Software
  - Built-in 4096x16 SRAM
  - Embedded PC Stack Level 16
- ◆ Rich DSP function
  - Hardware Circular Buffer support
  - MAC Computation power : 32 MIPS (max.)
  - Multi-Function Support: In MAC calculation, simultaneously access two operands from memory in one cycle
  - Extend Dynamic Range: A 40-bit accumulator to ensure in 512 successive multiple+additions no overflows
- ◆ Embedded Flash 512Kx16
  - Typical 1,000 erase/program cycles
  - Greater than 10 years Data Retention
- ◆ Software-based audio processing technical
  - Subband , ADPCM , Melody
- ◆ Support 24+2(ICE PAD can be as I/O) general purpose I/O port.
- ◆ Mono 16bit PWM or 16bit DAC
- ◆ 6 IRQ include 2 external interrupt
- ◆ SPI Master interface
- ◆ Three timers: Timer1, Timer2, RTC timer
- ◆ Support Spread Spectrum clocking to reduce EMI.
- ◆ Watch dog timer (WDT)
- ◆ Low voltage reset (LVR)
- ◆ PB0, PB1, PB2, PB3 support two edge modes for wake-up function are rising and falling edge trigger.
- ◆ 2 channel comparators and a two stage OPA with class AB output
- ◆ Microphone with AGC(Auto Gain Control)
- ◆ IR
- ◆ Touch with BG Pre-Charge
- ◆ Low voltage detector
- ◆ **Notice: VCC Decoupling Cap 10uF should be close to IC within 0.5cm.**
- ◆ **Notice: VCC Decoupling Cap 0.1uF should be close to IC within 0.5cm.**



## CST1608A(C) 16-bit Speech Processor

- ◆ **Notice: The width of VCC and VSS power line should be greater than 30 mil in PCB layout.**
- ◆ **Notice: Special I/O PortC2 will cause LED flashing light when power on.**
- ◆ **PortA4 and PortA5 are not supported touch function.**
  
- ◆ **Notice: PortC2 / PortC6 / PortC7 are forbidden to be directly connected to inductive components. For example, motors and coils, etc.**
- ◆ **Notice: PWM / DAC are forbidden to be connected to 4Ω speaker.**
  
- ◆ **CST16512A 和 CST1608A(C) 的差異如下:**
  - CST1608A(C) 的 I/O 增加 PortA4 和 PortA5。
  - CST1608A(C) 的 Touch 功能支援 BG Pre-Charge
  - CST1608A(C) 的 Microphone 增加 PGA gain boost(GBB)
  - CST1608A(C) 的 LVR 改為 8 種不同電壓，1.7V / 1.8V / 1.9V / 2.0V / 2.1V / 2.2V / 2.3V / 2.4V。
  - CST1608A(C) 的電壓工作範圍 1.8V~5.5V
  - CST1608A(C) 不支援 Deep sleep 功能
  - CST1608A(C) 的 PWM 可以調整趨動電流
  - CST1608A(C) 的 PortC2 和 PortC6 和 PortC7 禁止接電感性元件，例如馬達、線圈。
  - CST1608A(C)的 PWM / DAC 輸出 禁止接 4Ω 喇叭。
- ◆ CST1608A(C)的 PortA4 和 PortA5 不支援 Touch 功能
- ◆ 因為 CST16512A 沒有 PortA4 和 PortA5，所以如果從 CST16512A 的程式直接轉到 CST1608A(C)，需要注意 PortA4 和 PortA5 的設定。例如進入 sleep mode 時，要把 PortA4 和 PortA5 設定為 input 加 pull down R，以免漏電。



## CST1608A(C) 16-bit Speech Processor

### 3. CST1608A(C) Application Field

- MCU Application
- Electronic Dictionary
- Handheld Games
- Electronic Learning Aid (ELA)
- Electronics storybook

### 4. CST1608A(C) Block Diagram

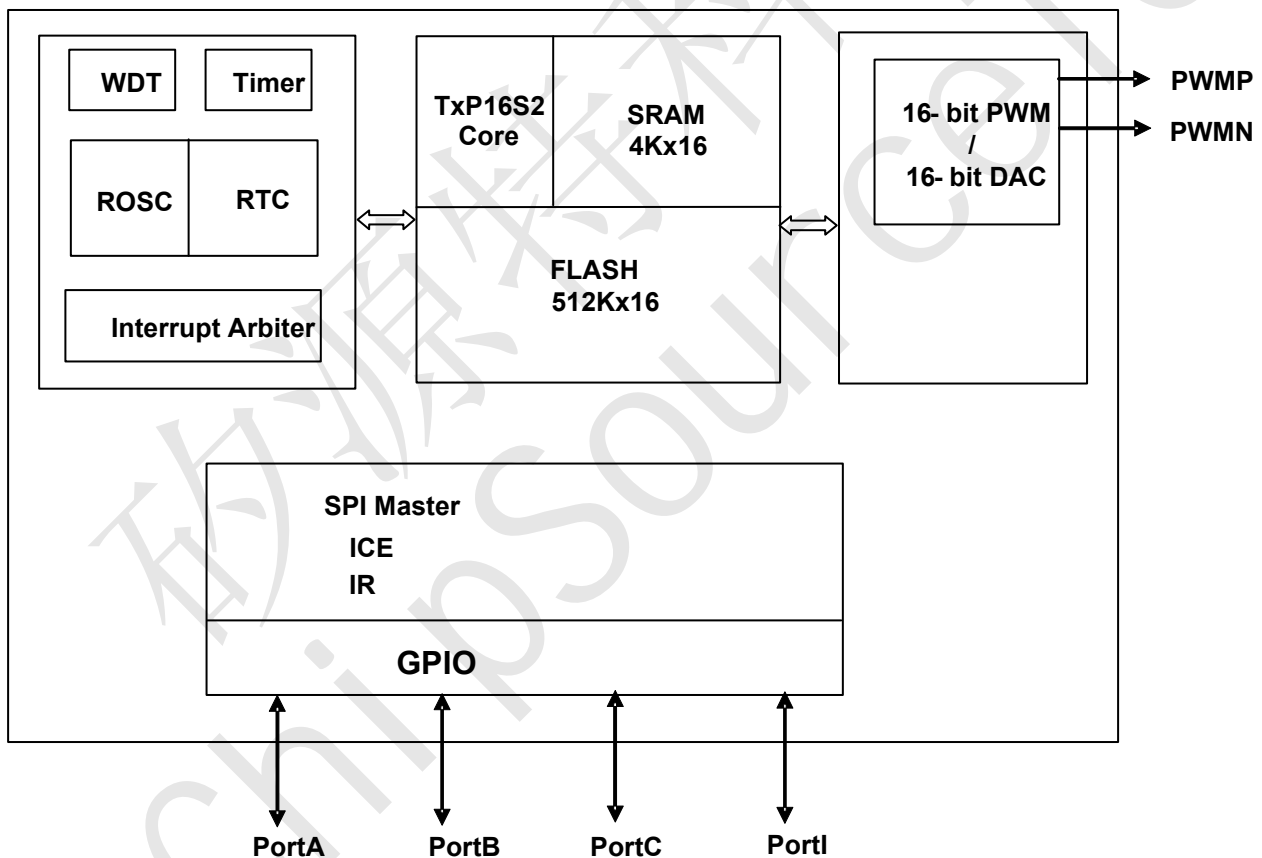


Figure 4.1



## CST1608A(C) 16-bit Speech Processor

### 4.1 Pin Assignments /Description

Pin Name	I/O	State after RESET	FUNCTIONS
Chip Power			
VCC	I	High	Chip Power Input
VSS	I	Low	Digital Ground
VPD	I	High	PWM IO PAD Power Input
VPS	I	Low	PWM IO PAD Ground
General Purpose I/O Port			
PortA[5:0]	I/O	Low	PortA is programmable Input/Output port
PortB[5:0]	I/O	Low	PortB is programmable Input/Output port
PortC[11:0]	I/O	Low	PortC is programmable Input/Output port PortC[2] is programmable Input/Output port when not connected to ICE ICE_VPP: PortC[2] is embedded ICE VPP pin when connected to ICE Probe.
PortI[1]/ ICE_SCLK	I/O	Low	PortI[1] is programmable Input/Output port when not connected to ICE Probe. Internal pull-down R (50K ohm) will be enable at ICE mode. ICE_SCLK: PortI[1] is embedded ICE clock pin when connected to ICE Probe.
PortI[0]/ ICE_SD	I/O	Low	PortI[0] is programmable Input/Output port when not connected to ICE Probe. Internal pull-down R (50K ohm) will be enable at ICE mode. ICE_SD: PortI[0] is embedded ICE data pin when connected to ICE Probe.
PWM Audio			
PWMP	O	Low	Digital PWM output(+) / Analog DAC output(+)
PWMN	O	Low	Digital PWM output(-) / Analog DAC output(-)

- ◆ **Notice: Special I/O PortC2 will cause LED flashing light when power on.**
- ◆ **Notice: PortC2 are forbidden to be connected with VCC or pull-up resistance.**



## CST1608A(C) 16-bit Speech Processor

### 5. CST1608A(C) Function Descriptions

#### 5.1 TxP16S2

As shown in the block diagram in Figure 4.1, the TxP16S2 is a 16-bit data width processing capability and all instructions are operated in one cycle except parameter data ROM(PM) access. The TxP16S2 not only provides general arithmetic such as addition, subtraction, shifter, normalize, and other logical operations, but it also involves circular buffer operations for complexity digital signal processing.

#### 5.2 TxP16S2 Registers

The TxP16S2 contains of register files are illustrated below:

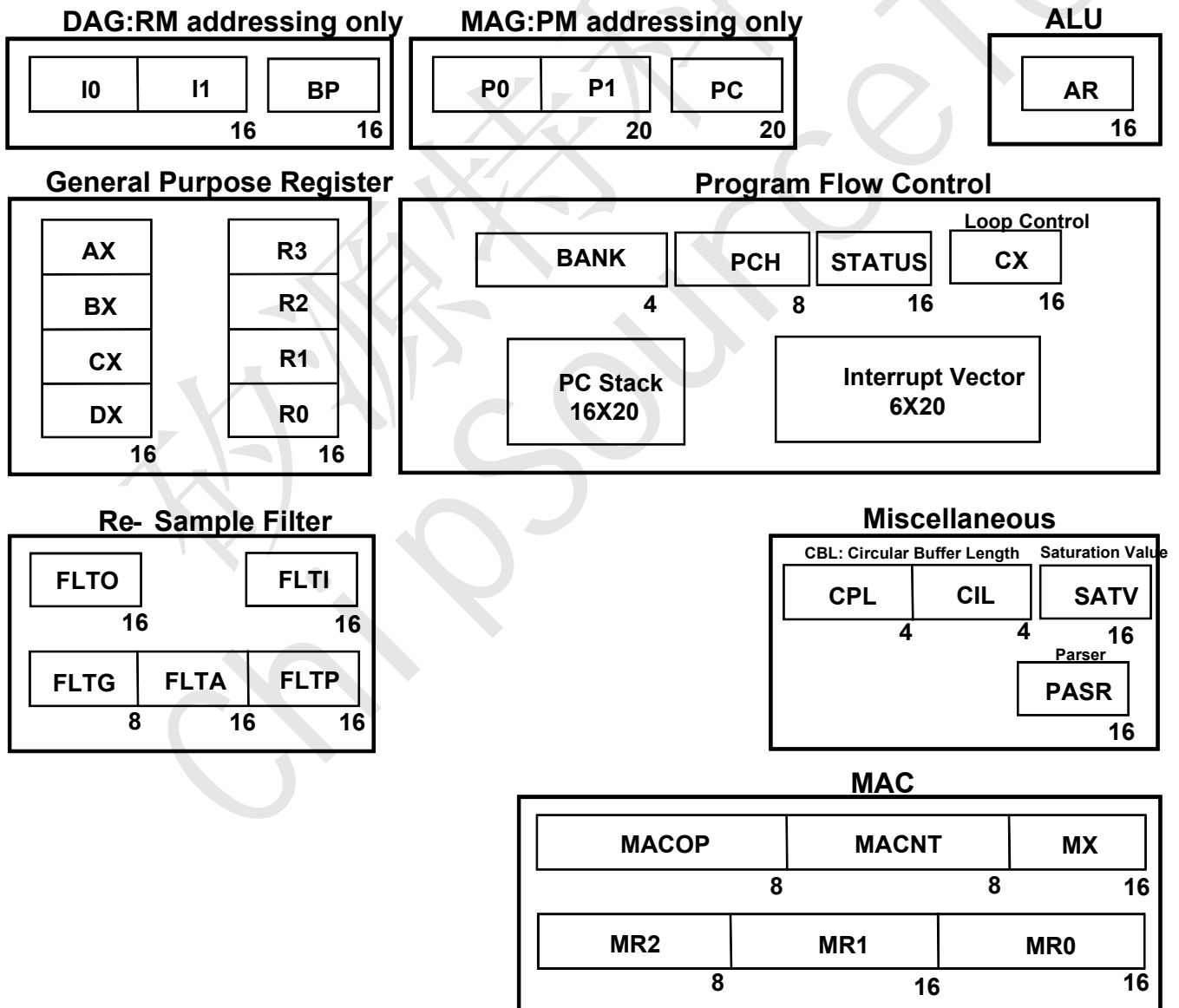


Figure 5.1 TxP16S2 Processor Core Registers



### CST1608A(C) 16-bit Speech Processor

#### REGISTER FILES DEFINE:

- |                                   |                                      |
|-----------------------------------|--------------------------------------|
| AR: Accumulator Register          | MR2: MUL/MAC Result Register 2       |
| I0: Index 0 Register              | AX: General AX Register              |
| I1: Index 1 Register              | BX: General BX Register              |
| BP: Base Pointer Register         | CX: General CX Register              |
| P0: Pointer 0 Register            | DX: General DX Register              |
| P1: Pointer 1 Register            | R0: General R0 Register              |
| MACOP: MAC Operation Register     | R1: General R1 Register              |
| MACNT: MAC Operation Loop Counter | R2: General R2 Register              |
| MX: MUL/MAC Input X Register      | R3: General R3 Register              |
| MR0: MUL/MAC Result Register 0    | CBL: Circular Buffer Length Register |
| MR1: MUL/MAC Result Register 1    | PASR: Parser Register                |

#### 5.2.1 Special Registers

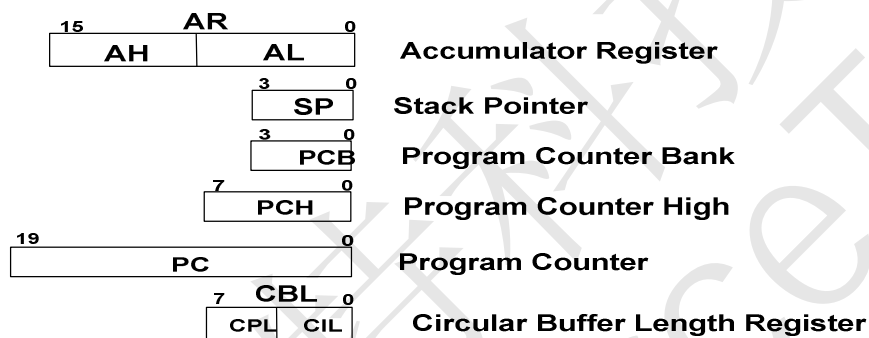


Figure 5.2 TxP16S2 Special Registers

- **Accumulator Register**  
The AR is a general-purpose 16-bit register that stores the result of last arithmetic or logical operation. In addition, any data write to AR will affect the status flag.
- **Stack Pointer**  
The SP is a 4-bit register that is for addressing Stack position. The SP will automatically increment / decrement
- **Program Counter Bank**  
The program memory map is divided into 16 banks by PCB register (Program Counter Bank). From BANK8 to BANK14 are system reserved. From BANK0 to BANK7 are implemented as Flash memory. The BANK15 is implemented as static RAM.
- **Program Counter High**  
The instruction “LJMP” and “LCALL” will refer PCH and PCB registers to compose of 20-bit pointer provides the 16x64K words PM addressing range.
- **Program Counter**  
The 20-bit PC register provides 16x64K-word addressing capability. It is responsible for MCU fetch now executing instruction.
- **Circular Buffer Length Register**  
Many algorithms such as convolution, correlation, and digital filter require the circular data buffers. The TxP16S2 supports circular buffer operating via the I0 vs. CIL and P0 vs. CPL. The modulus logic implements automatic modulus addressing for accessing RM/PM circular buffer data.



### CST1608A(C) 16-bit Speech Processor

#### 5.2.2 Common I/O Registers

The TxP16S2 involves 32 common I/O registers are shown in Table 5.1. There are defined the peripheral IO control and system register.

Symbol	Adr	Reset	RW	B15/b7	b14/b6	b13/b5	b12/b4	B11/b3	B10/b2	b9/b1	b8/b0	Description
STATUS.L	00H	20	R/W	INTEN	-	-	AQ	AN	AV	AC	AZ	System Status Flag
STATUS.H	00H	00	R/W	PA	FA	IntVWR	-	-	-	-	SPIM EN	
INTENA.L	01H	00	R/W	-	-	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0	Interrupt Enable
INTENA.H	01H	00	R/W	-	-	-	-	-	-	-	-	
INTREQ.L	02H	00	R/W	-	-	Req5	Req4	Req3	Req2	Req1	Req0	Interrupt Request
INTREQ.H	02H	00	R/W	-	-	-	-	-	-	-	-	
IntVect	03H	XX	R/W	IntVect[5:0]							Interrupt Vector access Window	
IOC_PA	04H	00	R/W	IOC_PA[5:0]							"1" = out, "0" = in of related PA bit	
IOC_PB	05H	00	R/W	IOC_PB[5:0]							"1" = out, "0" = in of related PB bit	
IOC_PCI.L	06H	00	R/W	IOC_PC[11:0]							"1" = out, "0" = in of related PC bit	
IOC_PCI.H	06H	00	R/W	IOC_PI[1:0]								"1" = out, "0" = in of related PI bit
PortA	07H	XX	R/W	PortA[5:0]							Read: in port Write: out port	
PortB	08H	XX	R/W	PortB[5:0]							Read: in port Write: out port	
PortCI.L	09H	XX	R/W	PortC[11:0]							Read: in port Write: out port	
PortCI.H	09H	XX	R/W	PortI[1:0]								Read: in port Write: out port
INTMASK.L	0AH	00	R/W	-	-	Mask5	Mask4	Mask3	Mask2	Mask1	Mask0	Interrupt Mask
INTMASK.H	0AH	00	R/W	-	-	-	-	-	-	-	-	
Reserve	0BH	XX	W									
Reserve	0CH	XX	XX									
Reserve	0DH	XX	XX									
Reserve	0EH	XX	XX									
Reserve	0FH	XX	XX									
Reserve	10H	XX	XX									
Reserve	11H	XX	XX									
Reserve	12H	XX	XX									
Reserve	13H	XX	XX									
Reserve	14H	XX	XX									
Reserve	15H	XX	XX									
Audio-Audio-DAC	16H	XX	W	Audio-PWM / Audio-DAC[15:0]							PWM / DAC Audio Channel	
Reserve	17H	XX	W									
MISC5.L	18H	00	R/W	THO	LVDO	CMP2_TIMER2	CMP1_TIMER1	CMP2_I_NT_INV	CMP1_I_NT_INV	CMP2_INT_EN	CMP1_INT_EN	System Miscellanea register #5
MISC5.H	18H	00	R/W	InterPLY_FREQ_DIV								
MISC4.L	19H	00	R/W	TOUCH_REALT	TINT_EN	TIS		SVT	THEN	PREC	-	System Miscellanea register #4
MISC4.H	19H	00	R/W	DAC_Bypass	DAC_OG			InterPLY_EN	Tone_Mode	InterPLY_FUN		
MISC3.L	1AH	00	R/W	D2ENB	LVDEN	PC7_ALG	PC6_ALG	PC5_ALG	PC4_ALG	PC2_ALG	PA0_ALG	System Miscellanea register #3
MISC3.H	1AH	00	R/W	CMPO2	CMPO1	LVD						
MISC2.L	1BH	00	R/W	CMP2E_N	CMP1E_N	OFT_SEL		GBB	-			System Miscellanea register #2
MISC2.H	1BH	00	R/W	ENAVS	BUZEN	PGAEN	0	RECEN	OPEN	0	DISPUR	
MISC1.L	1CH	00	R/W	TCS	DAC_EN	PWM_MUTE	DAC_MUTE_B	RC_RST	EXRST	LVR	WDT	System Miscellanea register #1
MISC1.H	1CH	00	R/W	CLR_RealT	-	-	RTC_WAKE	IR_EN	0	PWM_EN	MODX	
RealT	1DH	XX	R	RealT[15:0]							RealT register	
ClrWDT	1DH	XX	W									Clear WDT
IOP_IX	1EH	XX	W	IOPIX[7:0]							Programming IO Port index	
IOP_DAT	1FH	XX	W	IOPD[15:0]							Programming IO Port Data	

Table 5.1 Common I/O registers



### CST1608A(C) 16-bit Speech Processor

#### 5.2.3 Basic System Registers

##### ◆STATUS register

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
STATUS.L	00H	20	R/W	INTEN	-	-	AQ	AN	AV	AC	AZ	System Status Flag
STATUS.H	00H	00	R/W	PA	FA	IntVWR	-	-	-	-	SPIM_EN	

The Status register provides two main functions, the first system flag holds the status information generated by the computational blocks of the TxP16S2, which used for program sequencer control. The second indicated that special function of hardware module is enable or not.

For program flow control:

AZ	ALU or AR Result Zero
AN	ALU or AR Result Negative
AV	ALU Overflow
AC	ALU Carry
PA	Parser Queue available(Read only)
FA	Filter buffer available(Read only)

System hardware control:

INTEN	System global interrupt control bit
IntVWR	Interrupt Vector Table access window control bit
SPIM_EN	SPI master interface control bit

- ◆ Address 01H, 02H and 0AH: Interrupt control registers, the detail are illustrated in Interrupt section.
- ◆ Address 04H~15H: GPIO registers, the detail are illustrated in GPIO section.
- ◆ Address 16H: Audio-PWM control registers, the detail are illustrated in AUDIO section.

##### ◆System Miscellanea register #1

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC1.L	1CH	00	R/W	TCS	DAC_EN	PWM_MUTE	DAC_MUTE_B	RC_RST	EXRST	LVR	WDT	System miscellanea register #1
MISC1.H	1CH	00	R/W	CLR_RealT	-	-	RTC_WAKE	IR_EN	0	PWM_EN	MODX	

Clr_RealT	Set high to clear 32-bit RealT timer. (this bit only for TCS=1)
RTC_WAKE	RTC Wake Up Flag (Read Only)
IR_EN	IR 38K Enable
PWM_EN	Audio PWM enable
MODX	MODX=0 is chosen narrowband sound-effect filter. MODX =1 is chosen wideband sound-effect filter. Actual bandwidth is dependent on source signal sample-rate.
TCS	Enable RealT Timer
DAC_EN	Audio DAC enable
PWM_MUTE	Audio PWM mute enable
DAC_MUTE_B	Audio DAC mute enable
RC_RST	System Reset from internal RC reset



### CST1608A(C) 16-bit Speech Processor

EXRST	System Reset from external reset pin
LVR	System Reset from low voltage reset
WDT	System Reset from watch dog reset

Note1: RC\_RST, EXRST, LVR will clear WDT bits, except for WDT is set to 1. All of the reset status bits can be set or cleared in software. Setting one of these bits in software does not cause a device reset.

Note2: If programmer read the “RealT” register, it can get 32-bit timer based on 30.517ns(32.768MHz). An example is shown as follows. Address 1DH: RealT register.

```
io[RealT] = ar          ; write to reset the state machine of 32-bit real timer.
ar         = io[RealT]  ; read low-word timer[15:0]
ar         = io[RealT]  ; read high-word timer[31:16]
```

#### ◆ System Miscellanea register #2

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC2.L	1BH	00	R/W	CMP2EN	CMP1EN	OFT_SEL	GBB	-	-	-	-	System miscellanea register #2
MISC2.H	1BH	00	R/W	ENAVS	BUZEN	PGAEN	0	RECEN	OPEN	0	DISPUR	

ENAVS	Mic OP grounding to VPS
BUZEN	Default:0 , AGC enable
PGAEN	Default:0 , Programmable Gain Amplifier
RECEN	Default:0 , Microphone Enable
OPEN	Default:0 ,
DISPUR	Default:0 ,
CMP2EN	Default:0 , Comparator 2 Enable
CMP1EN	Default:0 , Comparator 1 Enable
OFT_SEL	Default:0 , Microphone Gain 0:50, 1:100, 2:150, 3:200
GBB	Default:0 , GBB=0 for PGA gain boost

#### ◆ System Miscellanea register #3

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC3.L	1AH	00	R/W	D2ENB	LVDEN	PC7_ALG	PC6_ALG	PC5_ALG	PC4_ALG	PC2_ALG	PA0_ALG	System miscellanea register #3
MISC3.H	1AH	00	R/W	CMPO2	CMPO1	-	-	-	-	-	-	

CMPO2	Comparator 2 Output (Read Only)
CMPO1	Comparator 1 Output (Read Only)
LVD	Default:0 , LVD voltage 1.8~ 2.98V for D2ENB=0(each step~ 40mV), 2.7V~4.47V for D2ENB=1(each step~ 60mV)
D2ENB	Default:0 , VCC divide 2 for LVD input
LVDEN	Default:0 , LVD Enable
PC7_ALG	Set PortC7 to Analog PIN
PC6_ALG	Set PortC6 to Analog PIN
PC5_ALG	Set PortC5 to Analog PIN
PC4_ALG	Set PortC4 to Analog PIN
PC2_ALG	Set PortC2 to Analog PIN
PA0_ALG	Set PortA0 to Analog PIN



### CST1608A(C) 16-bit Speech Processor

#### ◆ System Miscellanea register #4

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC4.L	19H	00	R/W	TOUCH_REALT	TINT_EN	TIS		SVT	THEN	PREC	-	System miscellanea register #4
MISC4.H	19H	00	R/W	DAC_Bypass	DAC_OG		InterPLY_EN	Tone_Mode	InterPLY_FUN			

DAC_Bypass	Bypass DAC input function	0:disable(default) 1:enable
DAC_OG	Default:0, DAC output gain = 1	
InterPLY_EN	0:DAC Interpolation disable	1: DAC Interpolation enable
Tone_Mode	0:TONE disable	1:TONE enable
InterPLY_FUN	0:DAC Interpolation disable	1: DAC Interpolation 8 point 2:DAC Interpolation 16 point 3: DAC Interpolation 32 point
TOUCH_REALT	Touch with Real-T function	0:disable & clear real-T stop flag 1:enable
TINT_EN	Touch INT Enable	
TIS	Touch I/O Port input select , 00: NA, 01:PortA, 10: NA, 11:PortC	
SVT	Touch SVT , 0: 1/4*VCC, 1: 1/2*VCC	
THEN	Touch Enable	
PREC	Touch Pre-charge 1.5V	

#### ◆ System Miscellanea register #5

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC5.L	18H	00	R/W	THO	LVDO	CMP2_TIMER2	CMP1_TIMER1	CMP2_INT_INV	CMP1_INT_INV	CMP2_INT_EN	CMP1_INT_EN	System Miscellanea register #5
MISC5.H	18H	00	R/W	InterPLY_FREQ_DIV								

InterPLY_FREQ_DIV	System Frequency Divisor for DAC interpolation, Default:0	
THO	TOUCH Output (Read Only)	
LVDO	LVD Output (Read Only)	
CMP2_TIMER2	Comparator 2 with Timer2 function	0:disable & clear Timer2 stop flag 1:enable
CMP1_TIMER1	Comparator 1 with Timer1 function	0:disable & clear Timer1 stop flag 1:enable
CMP2_INT_INV	Comparator 2 INT Inverse	
CMP1_INT_INV	Comparator 1 INT Inverse	
CMP2_INT_EN	Comparator 2 INT Enable	
CMP1_INT_EN	Comparator 1 INT Enable	



### CST1608A(C) 16-bit Speech Processor

#### ◆ Virtual Programming IO

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description	
VIO_IX	1EH	XX	W	-	-	-	IOPIX4	IOPIX3	IOPIX2	IOPIX1	IOPIX0	Programming IO Port index	
VIO_DATA	1FH	XX	W	IOPD[15:0]									Programming IO Port Data

Table 5.2 Virtual Programming IO

The operation steps of these group register, first select virtual IO port index then write data to programming IO port.

#### Virtual Programming IO Port

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	B12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description	
Timer1	00H	00	W	Timer0									Timer1
Timer1	00H	00	R	Current Timer1 Value									
Timer2	01H	00	W	Timer1									Timer2
Timer2	01H	00	R	Current Timer2 Value									
RTCTimer	02H	00	RW	RTCTimer									RTC Timer
FetchCNT	03H	00	W	-	-	-	-	-	-	Fetch Timer2	Fetch Timer1	Fetch Timer1/2 Current Value	
Reserve	04H	-	-										
PA_PD50K	05H	00	W	PA_PD50K[5:0]									PortA Pull Down 50K Enable
PB_PD50K	06H	00	W	PB_PD50K[5:0]									PortB Pull Down 50K Enable
PCI_PD50K.L	07H	00	W	PC_PD50K[7:0]									PortCI Pull Down 50K Enable
PCI_PD50K.H	07H	C0	W	PI_PD50K[1:0]	-	-	PC_PD50K[11:8]					PortCI Pull Down 50K Enable <b>(PortI default : Enable PD50K)</b>	
PA_PD220K	08H	00	W	PA_PD220K[5:0]									PortA Pull Down 220K Enable
PB_PD220K	09H	00	W	PB_PD220K[5:0]									PortB Pull Down 220K Enable
PCI_PD220K.L	0AH	00	W	PC_PD220K[7:0]									PortCI Pull Down 220K Enable
PCI_PD220K.H	0AH	00	W	PI_PD220K[1:0]	-	-	PC_PD220K[11:8]					PortCI Pull Down 220K Enable	
PA_PD1M	0BH	00	W	PA_PD1M[5:0]									PortA Pull Down 1M Enable
PB_PD1M	0CH	00	W	PB_PD1M[5:0]									PortB Pull Down 1M Enable
PCI_PD1M.L	0DH	00	W	PC_PD1M[7:0]									PortCI Pull Down 1M Enable
PCI_PD1M.H	0DH	00	W	PI_PD1M[1:0]	-	-	PC_PD1M[11:8]					PortCI Pull Down 1M Enable	
PA_TOUCH	0EH	00	W	PA_TOUCH[5:0]									PortA Touch Enable
PB_TOUCH	0FH	00	W	PB_TOUCH[5:0]									PortB Touch Enable
PCI_TOUCH.L	10H	00	W	PC_TOUCH[7:0]									PortCI Touch Enable
PCI_TOUCH.H	10H	00	W	PI_TOUCH[1:0]	-	-	PC_TOUCH[11:8]					PortCI Touch Enable	
Reserve	11H	00	W										
Reserve	12H	00	W										
Reserve	13H	00	W										
WAKEN_PA	14H	00	W	PortA_WAKEN[5:0]									PortA WAKE UP Enable
WAKEN_PB	15H	00	W	PortB_WAKEN[5:0]									PortB WAKE UP Enable
WAKEN_PC	16H	00	W	PortC_WAKEN[11:0]									PortC WAKE UP Enable
WAKELV_PA	17H	00	W	PortA_WAKELV[5:0]									PortA WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PA bit
WAKELV_PB	18H	00	W	PortB_WAKELV[5:0]									PortB WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PB bit
WAKELV_PC	19H	00	W	PortC_WAKELV[11:0]									PortC WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PC bit
WAKEDLV_PB	1AH	00	W	PortB_WAKEDLV[3:0]									PortB Double-Enable
Reserve	1BH	01	W										
Reserve	1BH	00	W										
Reserve	1BH	00	R										
Reserve	1BH	00	R										
Reserve	1CH	XX	W										
Reserve	1CH	XX	R										



## CST1608A(C) 16-bit Speech Processor

### 5.3 PC Stack

The PC STACK is TxP16S2 special embedded memory used to save (PC+1) value, which is composed with 16-level.

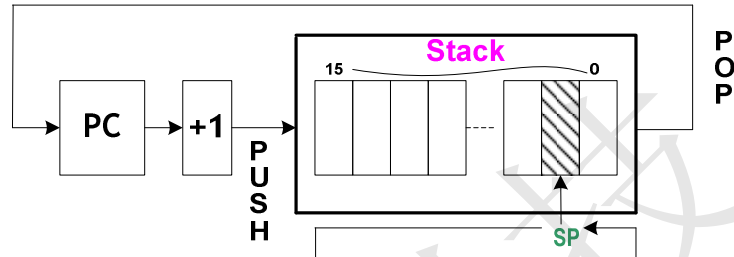


Figure 5.3 PC Stack Structure

Stack's top value is indexed by stack pointer (SP) register. When CALL instruction is executed, then the (PC+1) will PUSH onto stack addressing by SP and it will auto decrement. At the end of subroutine when RETS instruction is executed the SP will auto increment and stack content of pointer by SP will POP into PC.

The contents of STACK and SP are neither readable nor writeable by instruction. The SP is initialized to "0" after RESET.



### CST1608A(C) 16-bit Speech Processor

#### 5.4 Interrupt

##### 5.4.1 Interrupt Vector Table

The Interrupt Vector Table is TxP16S2 special embedded memory, which is composed with 6-level of FIFO, used to store the index of interrupt service routine (ISR) address. User can access Interrupt Vector Table by read/write IntVect I/O register, which refers PCB register to compose of 20-bit address.

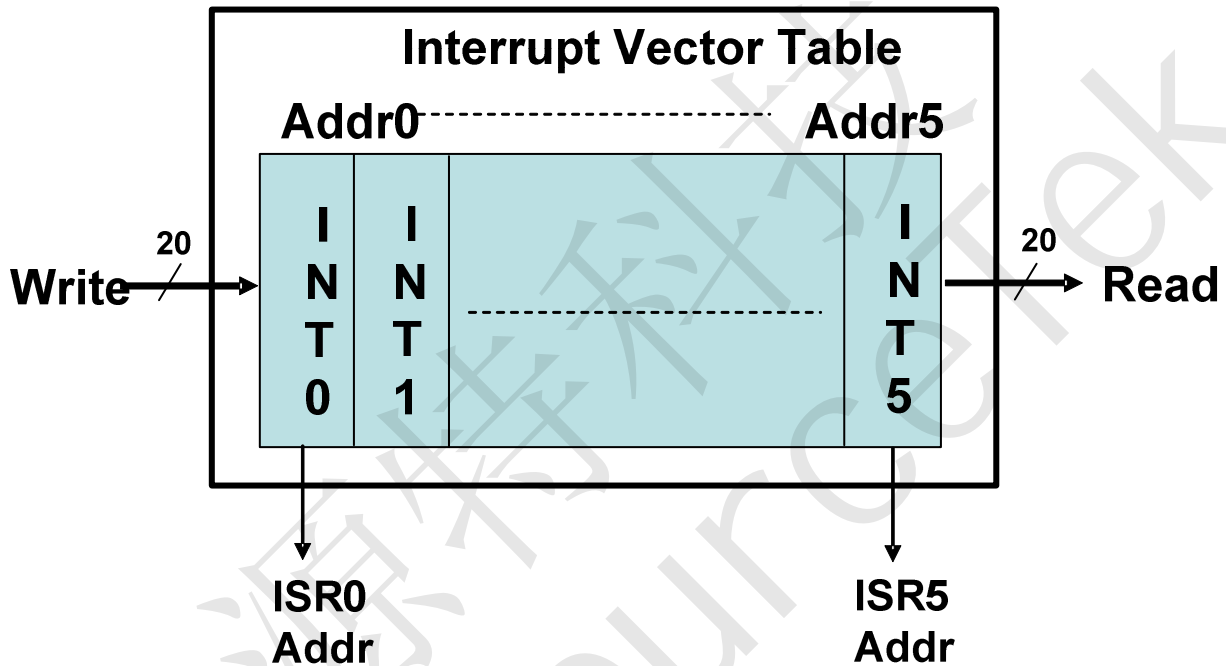


Figure 5.4 Interrupt Vector Structure

##### 5.4.2 Interrupt Controller

###### Common I/O registers

Symbol	Adr	Reset	RW	B7	B6	B5	B4	B3	B2	B1	B0	Description
STATUS	00H	00	R/W	INTEN	-	-	-	AN	AV	AC	AZ	System Status Flag
INTENA.L	01H	00	R/W	-	-	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0	Interrupt Enable
INTENA.H	01H	00	R/W	-	-	-	-	-	-	-	-	
INTREQ.L	02H	00	R/W	-	-	Req5	Req4	Req3	Req2	Req1	Req0	Interrupt Request
INTREQ.H	02H	00	R/W	-	-	-	-	-	-	-	-	
INTMASK.L	0AH	00	R/W	-	-	Mask5	Mask4	Mask3	Mask2	Mask1	Mask0	Interrupt Mask
INTMASK.H	0AH	00	R/W	-	-	-	-	-	-	-	-	

This chip provides several interrupt sources, including internal Audio PWM, Timer1, Timer2, RTC, SPI master, Flash ready, Touch, and 2 external ExtINT0, ExtINT1, interrupts. More details control will describe as follows:

Interrupt Source	Interrupt Vector	Priority
Audio PWM Timer	0H	INT0_IRQ (highest)
Timer1 / ExtINT0 / SPI Master <b>(Note)</b>	1H	INT1_IRQ
Timer2 / ExtINT1 / SPI Master <b>(Note)</b>	2H	INT2_IRQ
RTC Timer / SPI Master <b>(Note)</b>	3H	INT3_IRQ
Flash Erase/Program Ready <b>(Note)</b>	4H	INT4_IRQ
Touch / SPI Master <b>(Note)</b>	5H	INT5_IRQ (lowest)

Table 5.3 Interrupt Sources

**Note: Only one interrupt source can be selected for each interrupt vector.**



### CST1608A(C) 16-bit Speech Processor

(a) Global interrupt enable (INTEN)

The global interrupt INTEN controls the enable/disable of all interrupts. When INTEN is cleared to “0”, all interrupts are disabled. When INTEN is set to “1”, all interrupts are enabled (but still dependent on value of INTENA register). The INTEN is initialized to “0” after power on.

(b) Interrupt enable (INTENA)

The interrupt enable from ENA5 to ENA0 are shown in above. An interrupt is allowed when these control bit are set to “1”, and interrupt is inhibit when these control bit are cleared to “0”. They are all initialized to “0” after power on.

(c) Interrupt request (INTREQ)

If an interrupt raising edge request is generated, the related interrupt request bit is set to “1” by hardware and waits for interrupt accept. INTREQ can be cleared to “0” by software. Hardware will not clear this bit. INTREQ are all initialized to “0” after power on.

(d) Interrupt mask (INTMASK)

The interrupt can be masked by setting bit5~ bit0 of interrupt mask register as above. Each interrupt source in the system can be masked individually.

(e) Interrupt Priority

INT0\_IRQ (highest) > INT1\_IRQ > INT2\_IRQ > INT3\_IRQ > INT4\_IRQ > INT5\_IRQ (lowest)

#### 5.4.3 Interrupt Processing

When any interrupt request(INTREQ) is generated, the acceptance of interrupt is decided by the interrupt enable(ENA) and global interrupt enable(INTEN). If the global interrupt enable(INTEN), related interrupt enable bit(ENA) are set to “1” and related mask bit(MASK) are cleared to 0, that interrupt will be accepted on the next clock. These following procedures will automatically be done in one clock cycle by hardware showing below:

- (1) Program Counter(PC), PCB, PCH, AR and FLAG will be stored in special hardware registers.
- (2) PC will be set to the corresponding interrupt entry address by refer to interrupt vector table.
- (3) The global interrupt enable (INTEN) is cleared to “0”, which avoids the nest interrupt happened.

When interrupt service routine was finished, an RETI instruction will perform the procedures by hardware showing as follows:

- (1) Restore the stored PC, PCB, PCH, AR and FLAG.
- (2) The global interrupt enable (INTEN) is set to “1”, which allows to accept the subsequent interrupt.

Before executing RETI instruction, the corresponding interrupt request (INTREQ) bit must be cleared to “0” by software. If the request bit is not cleared, the same interrupt will be accepted again.



## CST1608A(C) 16-bit Speech Processor

### 5.5 MAC (16-bit X 16-bit Multiplier and Accumulator)

A 16 bit x 16 bit MAC is provided for digital signal processing. The core of MAC operation is multiply MX&MY with 2'S complement operand and accumulation previous 40-bit MF then rounding store result in the 40-bit MR register. The basic MAC architecture is shown as Figure 5.5.

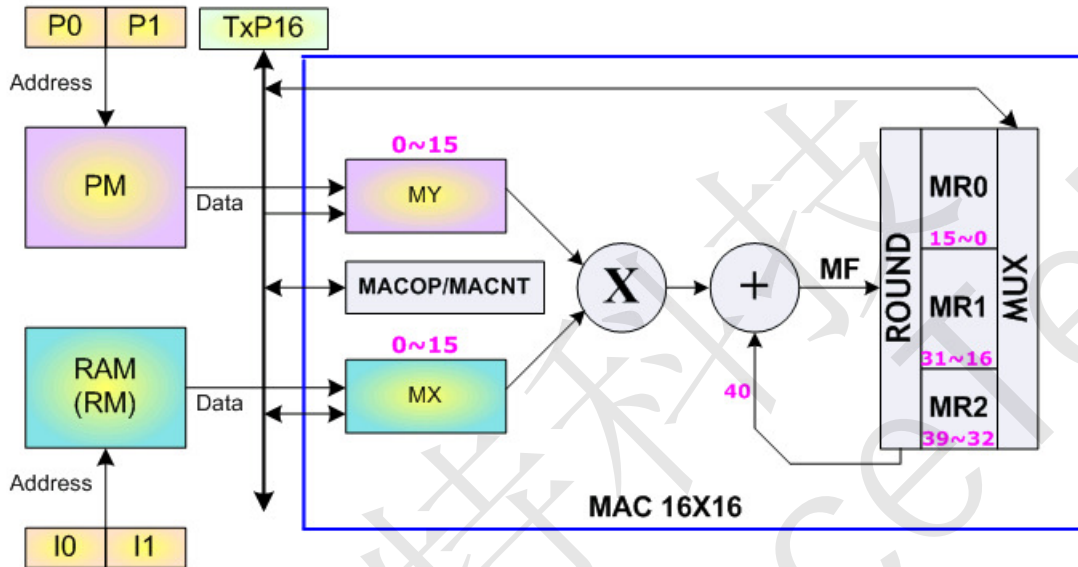


Figure 5.5 MAC Architecture

Define MAC module registers:

MX: MAC input 16-bit X register

MR: Multiplier or MAC result 40-bit register

MACOP: MAC operation define register

Symbol	Reset	RW	B7	B6	B5	B4	B3	B2	B1	B0	Description
MACOP	00	R/W	RND1	RND0	P0/P1	POP:+/-	I0/I1	IOP:+/-	MY:SU(01)		MAC Operation Setup

MACNT: MAC equation loop counter, max to 255

Basically, multiplier operates equation:

$$MR = MX * MY(SU) \rightarrow MX: \text{signed}, MY: \text{unsigned}$$

Permission MY is AR or immediate value (-128~127), MX and MY are signed or unsigned assign by MACOP. So, actual multiplier instruction likes this:

$$MR = MX * AR \quad \text{or} \quad MR = MX * 56$$

The operation of MAC equation is:

$$MR = MR + (MX * MY(SU)) \ll RND, MX = RM[I0++], MY = PM[P1--]$$

The means of equation is signed MX multiply unsigned MY the result value shift left RND bits and add previous MR then write back to MR.



## CST1608A(C) 16-bit Speech Processor

Simultaneously, load new value to   MX fixed from RM[index operation]  
  MY fixed from PM[pointer operation]

Therefore, MAC array operation like this:

$$MR = MR + ( MX*MY(SS) ) \ll 1, MX=RM[I0++], MY=PM[P1--]$$

$$MR = MR + ( MX*MY(SS) ) \ll 1, MX=RM[I0++], MY=PM[P1--]$$

·  
·  
·

$$MR = MR + ( MX*MY(SS) ) \ll 1, MX=RM[I0++], MY=PM[P1--]$$

Successive 64 times

Actual just one line of instruction present in assembly coding like this:

$$MACNT = 63$$

This is very benefit for reducing code size. Of course, we need setup MACOP register previous; at this example is like this;

Symbol	Reset	RW	B7	B6	B5	B4	B3	B2	B1	B0	Description
MACOP	00	R/W	RND1	RND0	P0/P1	POP:+/-	I0/I1	IOP:+/-	MY:SU(01)		MAC Operation Setup
			0	1	1	1	0	0	0	0	

*Note: Successive MAC operation will cause MCU interrupt disable.*



## CST1608A(C) 16-bit Speech Processor

### 6. CST1608A(C) Memory Configuration

#### 6.1 Internal Program/Parameter Memory

TxP16S2 consider both instruction and data ROM are the same as program ROM(PM), so it's very flexible and efficient for instruction and data memory allocation in PM. The total of logical PM space is 16 banks. Each bank has 64 K space. The 16 x 64K space is addressed by memory address generator unit (MAG). From BANK0 to BANK7 are implemented as Flash memory. From BANK8 to BANK14 are system reserved. The BANK15 is implemented as static RAM. More details control will describe as follows:

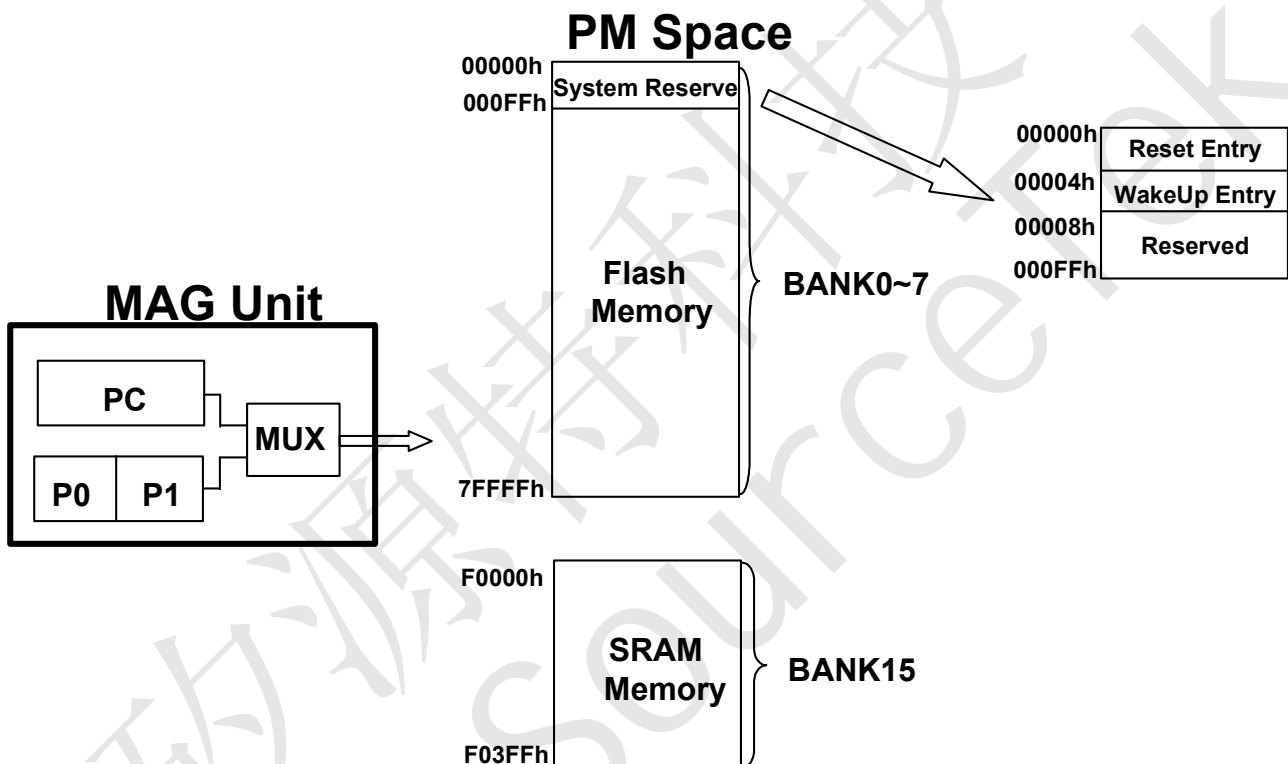


Figure 6.1 PM block diagram

When TxP16S2 executed an instruction, the PM address is generated from PC register. Similarity, when it access a word data, the PM address is composed with 20-bit from P0 or P1. System will auto adjust execute target space when program context switch between BANK0~7 and BANK15.

#### 6.2 Internal Data Working SRAM

The internal data working ram space is totally 4Kx16-bit that named as RM. Addressing ranged from 0x0000 through 0x0FFF, which is generated by Data Address Generator Unit (DAG). Note: Index register (I0,I1,BP) are 16-bit width, but RM address line is only 12-bit width(0~4095); therefore, RM address will be warping when index value exceed 4095.

*Note: Total SARM size is 4Kx16-bit. If PM(BANK15 / 1Kx16-bit) is used, RM will be 3Kx16-bit. If PM(BANK15) is no used, RM will be 4Kx16-bit. SARM configuration is set by option code.*



## CST1608A(C) 16-bit Speech Processor

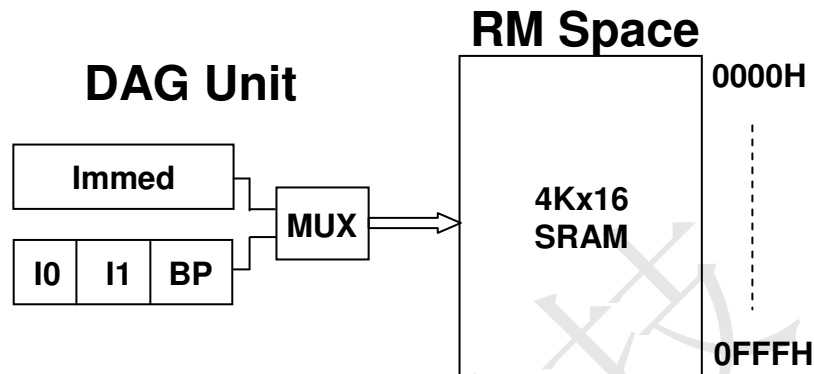


Figure 6.2 RM block diagram

### 6.3 Data Stack

A Last In First Out (LIFO) STACK is implementation for temporary data storage in RM memory. Generally, Data Stack is start-up at the bottom of RM, so BP is usually set to 0x0FFF.

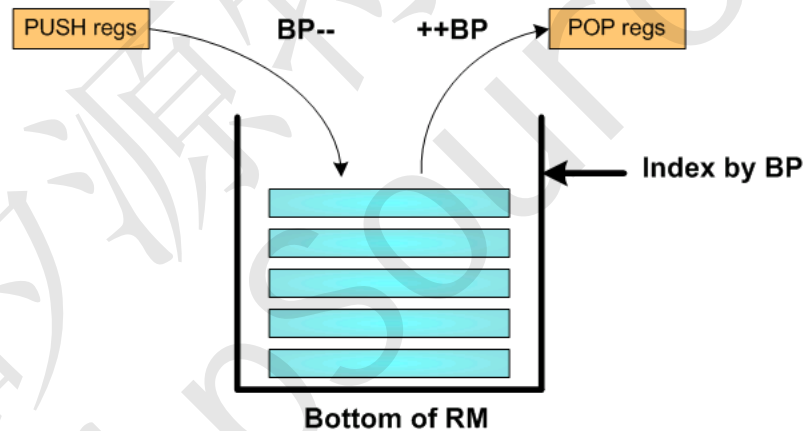


Figure 6.3 Data Stack Structure

Data Stack's top value is indexed by base pointer (BP) register. When PUSH instruction is executed, the "regs" will PUSH onto stack which address by BP and it will auto decrement. If POP instruction is performed, the BP will auto increment and stack content of pointed by BP will POP into "regs".



### CST1608A(C) 16-bit Speech Processor

## 7. CST1608A(C) Peripherals

### 7.1 Programmable Timers

Common I/O registers

Symbol	Adr	Reset	RW	B7	B6	B5	B4	B3	B2	B1	B0	Description
STATUS.L	00H	20	R/W	INTEN	-	-	AQ	AN	AV	AC	AZ	System Status Flag
STATUS.H	00H	00	R/W	PA	FA	IntVWR	-	-	-	-	SPIM EN	
INTENA.L	01H	00	R/W	-	-	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0	Interrupt Enable
INTENA.H	01H	00	R/W	-	-	-	-	-	-	-	-	
INTREQ.L	02H	00	R/W	-	-	Req5	Req4	Req3	Req2	Req1	Req0	Interrupt Request
INTREQ.H	02H	00	R/W	-	-	-	-	-	-	-	-	

Virtual Programming IO Port

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	B12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
Timer1	00H	00	W	Timer0								Timer1
Timer1	00H	00	R	Current Timer1 Value								
Timer2	01H	00	W	Timer1								Timer2
Timer2	01H	00	R	Current Timer2 Value								
RTCTimer	02H	00	RW	RTCTimer								RTC Timer
FetchCNT	03H	00	W	-	-	-	-	-	-	Fetch Timer2	Fetch Timer1	Fetch Timer1/2 Current Value

#### 7.1.1 Audio PWM Timer

Audio PWM timer is fixed generate 32kHz interrupt request when INTENA0 bit is turned on, *If Auto FIFO EN is setting the interrupt request rate =  $F_t / \text{FIFO Length}$ .* ***Notice: It should be avoided that INTENA bit0 is turned on/off quickly. If need to turned on/off INTENA bit0 quickly, recommend to use INTMASK bit0.***

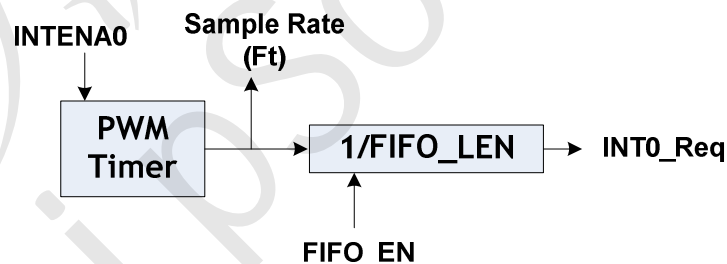


Figure 7.1 Audio PWM Timer Structure

#### 7.1.2 Timer1 & Timer2

The clock source of Timer1 & Timer2 comes from fixed 32.768Mhz or External Clock Input, It contains 16-bit write-only counter register. If Timer enable correspond with the INTENA bit is turned on then counting to time out, an interrupt request will be generated. At the same time, TnC will be reloaded into Timer register and up-count again. If the global interrupt enable, an interrupt signal is generated at the next clock.



## CST1608A(C) 16-bit Speech Processor

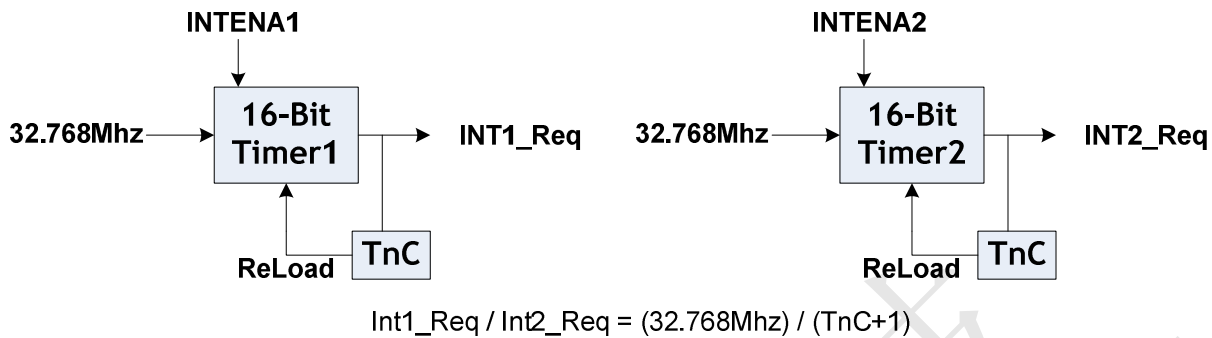


Figure 7.2 Timer1 & Timer2 Structure

### 7.1.3 RTC(Real Time Clock) Timer

The RTC Timer input frequency is Low power RC oscillator 32768Hz [-30%@1.8V](#) ~ +30%@5.5V (LP32K). It contains 16-bit counter register. RTC generates interrupt request or wake-up MCU when in halt mode or interrupt in normal mode. The wake-up function can be disabled by option.

**The frequency of  $Int3\_Req = 32768Hz / (RTC\ Timer\ TnC+1)$**

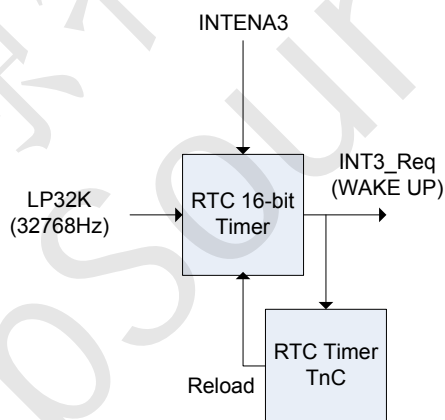


Figure 7.3 RTC Timer Structure



### CST1608A(C) 16-bit Speech Processor

#### 7.2 General Purpose I/O Ports

The CST1608A(C) provides 4 I/O ports for user application. There are four I/O port, PA0~PA5, PB0~PB5, PC0~PC11 and PI0~PI1. The input/output bits programmable by IOC control register respectively. PA0~PA5, PB0~PB5, PC0~PC11 wake-up function enable or disable by user program. PA1 and PA2 are external interrupt pins. The Pull-Down 50K or 220K or 1M resistor of each pin can be programmed by user program. **If I/O pin is set to output pin, the Pull-Down 50K and Pull-Down 1M resistor will be disable.**

The basic I/O schematic is showed in Figure 7.5.

#### Common I/O registers

Symbol	Adr	Reset	RW	B15/b7	b14/b6	b13/b5	b12/b4	B11/b3	B10/b2	b9/b1	b8/b0	Description
IOC_PA	04H	00	R/W	IOC_PA[5:0]				"1" = out, "0" = in of related PA bit				
IOC_PB	05H	00	R/W	IOC_PB[5:0]				"1" = out, "0" = in of related PB bit				
IOC_PCI.L	06H	00	R/W	IOC_PC[7:0]				"1" = out, "0" = in of related PC bit				
IOC_PCI.H	06H	00	R/W	IOC_PI[1:0]					IOC_PC[11:8]			"1" = out, "0" = in of related PI bit
PortA	07H	XX	R/W	PortA[5:0]				Read: in port Write: out port				
PortB	08H	XX	R/W	PortB[5:0]				Read: in port Write: out port				
PortCI.L	09H	XX	R/W	PortC[7:0]				Read: in port Write: out port				
PortCI.H	09H	XX	R/W	PortI[1:0]					PortC[11:8]			Read: in port Write: out port

#### Virtual Programming IO Port

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	B12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
PA_PD50K	05H	00	W	PA_PD50K[5:0]				PortA Pull Down 50K Enable				
PB_PD50K	06H	00	W	PB_PD50K[5:0]				PortB Pull Down 50K Enable				
PCI_PD50K.L	07H	00	W	PC_PD50K[7:0]				PortCI Pull Down 50K Enable				
PCI_PD50K.H	07H	C0	W	PI_PD50K[1:0]	-	-			PC_PD50K[11:8]			PortCI Pull Down 50K Enable <b>(PortI default : Enable PD50K)</b>
PA_PD220K	08H	00	W	PA_PD220K[5:0]				PortA Pull Down 220K Enable				
PB_PD220K	09H	00	W	PB_PD220K[5:0]				PortB Pull Down 220K Enable				
PCI_PD220K.L	0AH	00	W	PC_PD220K[7:0]				PortCI Pull Down 220K Enable				
PCI_PD220K.H	0AH	00	W	PI_PD220K[1:0]	-	-			PC_PD220K[11:8]			PortCI Pull Down 220K Enable
PA_PD1M	0BH	00	W	PA_PD1M[5:0]				PortA Pull Down 1M Enable				
PB_PD1M	0CH	00	W	PB_PD1M[5:0]				PortB Pull Down 1M Enable				
PCI_PD1M.L	0DH	00	W	PC_PD1M[7:0]				PortCI Pull Down 1M Enable				
PCI_PD1M.H	0DH	00	W	PI_PD1M[1:0]	-	-			PC_PD1M[11:8]			PortCI Pull Down 1M Enable
PA_TOUCH	0EH	00	W	PA_TOUCH[3:0]				PortA Touch Enable				
PB_TOUCH	0FH	00	W	PB_TOUCH[5:0]				PortB Touch Enable				
PCI_TOUCH.L	10H	00	W	PC_TOUCH[7:0]				PortCI Touch Enable				
PCI_TOUCH.H	10H	00	W	PI_TOUCH[1:0]	-	-			PC_TOUCH[11:8]			PortCI Touch Enable
Reserve	11H	00	W	-								
Reserve	12H	00	W	-								
Reserve	13H	00	W	-								
WAKEN_PA	14H	00	W	PortA_WAKEN[5:0]				PortA WAKE UP Enable				
WAKEN_PB	15H	00	W	PortB_WAKEN[5:0]				PortB WAKE UP Enable				
WAKEN_PC	16H	00	W	PortC_WAKEN[11:0]				PortC WAKE UP Enable				
WAKELV_PA	17H	00	W	PortA_WAKELV[5:0]				PortA WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PA bit				
WAKELV_PB	18H	00	W	PortB_WAKELV[5:0]				PortB WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PB bit				
WAKELV_PC	19H	00	W	PortC_WAKELV[11:0]				PortC WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge related PC bit				
WAKEDLV_PB	1AH	00	W	PortB_WAKEDLV[3:0]				PortB Double-Enable				

These totally 26 I/O pins work not only just a general input/output port function but also can be configured as SPI master, Analog input, IR 38K, External interrupt etc. For more detail please refer to relative section.



## CST1608A(C) 16-bit Speech Processor

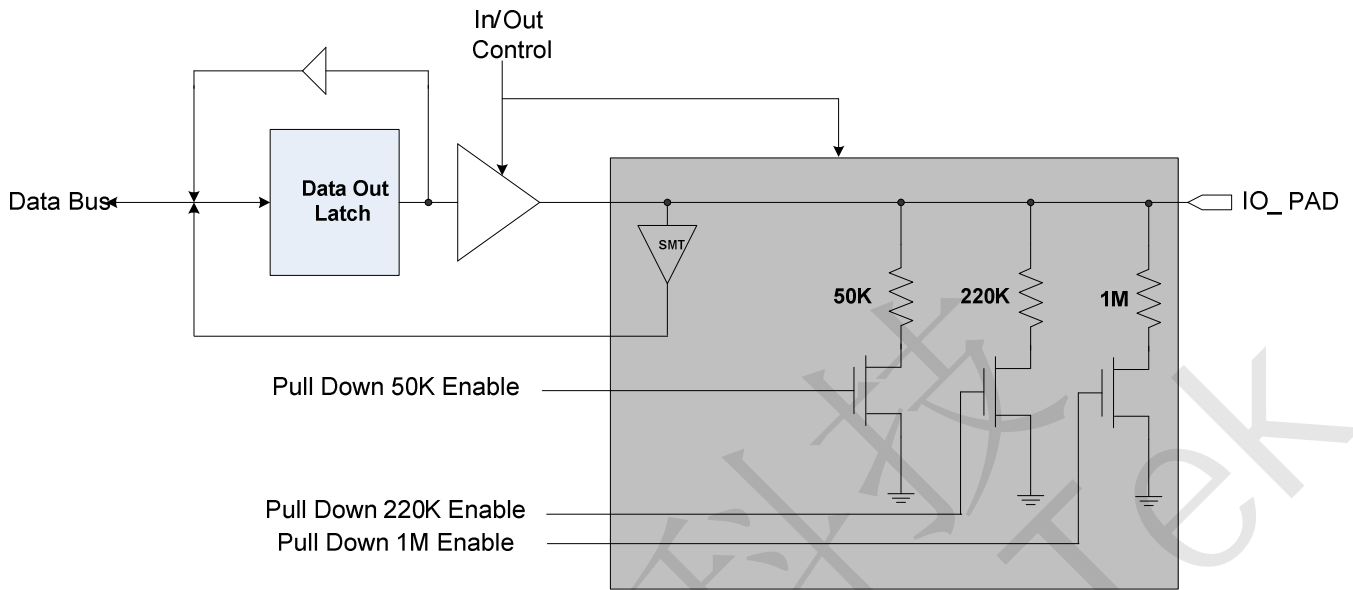


Figure 7.4 Basic I/O Configuration

### Virtual Programming IO Port

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	B12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
WAKEDLV_PB	1AH	00	W	PortB_WAKEDLV[3:0]							PortB Double-Enable	

PB0, PB1, PB2 and PB3 support two edge mode which is rising and falling edge trigger for wake-up function. The rising and falling edge trigger is selected by user program.

The PortC4, PortC5, PortC6, PortC7 also provides 2 comparators configuration for user application that each comparator enable or disable by program. The output of comparator is bit14 and bit15 of system miscellanea register #3.

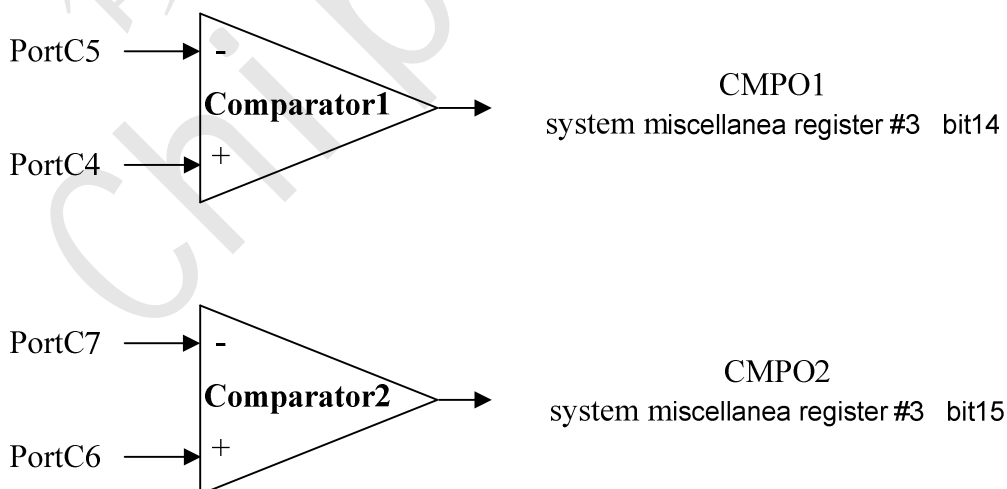


Figure 7.5 Comparator Configuration



### CST1608A(C) 16-bit Speech Processor

#### 7.3 Extension Device

CST1608A(C) built-in special hardwires for external device connection capability are listed below:

##### 7.3.1 SPI Master Controller

In order to enable SPI Master Controller interface, user should set STATUS.b8 = 1 before SPI Master Controller operation.

##### 7.3.1.1 Features

- Serial clock rate: 16.384 MHz / 8.192 MHz / 4.096 MHz
- Support four standard SPI modes
- Built in 1x16 bits data buffer

I/O Port	SPI interface	Direction	Description
PortB.2	SO	I/O	Serial Data output
PortB.1	SI	I/O	Serial Data input
PortB.0	SCK	O	Serial Clock
PortB.5	CS	O	Chip Select(free assign by user)

##### 7.3.1.2 Control/Data Registers

Symbol	Adr	Reset	RW	b15/b7	B14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
CTRL_W.L	55H	01	W	-	-	RCV	SEND	-	-	Total Byte		Write Control Register Low Byte
CTRL_W.H	55H	00	W	CPOL	CPHA	CKSEL		ICS	-	Which_INT		Write Control Register High Byte
CTRL_R.L	55H	01	R	DATOK	0	RCV	SEND	0	0	Total Byte		Read Control Register Low Byte
CTRL_R.H	55H	00	R	CPOL	CPHA	CKSEL		ICS	0	Which_INT		Read Control Register High Byte

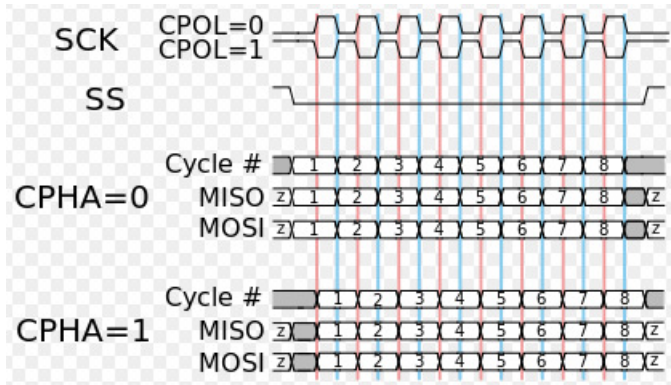
Total Send/Receive Byte	Default:1 , Total byte number of sending or receiving
SEND	Trigger sending data
RCV	Trigger receiving data.
Which_INT	Default:0 , assign SPI interrupt to which interrupt vector , 00:int1 01:int2 10:int3 11:int5
ICS	Internal SPI command select enable
CKSEL	00: N/A , 01: 16.384MHz , 10: 8.192 MHz , 11: 4.096 MHz
CPHA	Clock Phase
CPOL	Clock Polarity
DATOK	Indicate transmit/receive data O.K

#### SPI Modes

SPI Mode	Conditions	Leading Edge	Trailing eDge
0	CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)
1	CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)



### CST1608A(C) 16-bit Speech Processor



Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
DATA_W	56H	XX	W	DATA[15:0]							Write Transmission Data Value	
DATA_R	56H	XX	R	DATA[15:0]							Read Received Data Value	

DATA [15:0]: Transmit/Receive Data Value



## CST1608A(C) 16-bit Speech Processor

### 7.3.2 PortA3 for IR 38KHz Modulation

I/O Port	IR	Direction	Description
PortA.3	IR 38K	O	IR 38K Output

PA3 can provide 38 KHz modulation function. I/O port PA3 combine 38 KHz modulator with register DATA\_PA bit 3, this function is enabled by bit11 of system miscellanea register #1. If bit11 of system miscellanea register #1 is set to high, I/O port PA3 will output 38 KHz clock signal when PA3 is configured as output port and DATA\_PA bit3 = 1. In contrast, 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.

矽源特科技  
ChipSourceTek



### CST1608A(C) 16-bit Speech Processor

#### 7.3.3 Touch Controller

##### 7.3.3.1 Features

- Touch controller with one dedicated comparator
- Use Real-T counter to count the output pulse width of comparator
- Provides an interrupt for efficient programming.
- *PortA4 and PortA5 are not supported touch function.*

##### 7.3.3.2 Control Registers

###### ◆ System Miscellanea register #4

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC4.L	19H	00	R/W	TOUCH_REALT	TINT_EN	TIS		SVT	THEN	PREC	-	System miscellanea register #4
MISC4.H	19H	00	R/W	DAC_Bypass	DAC_OG		InterPLY_EN	Tone_Mode	InterPLY_FUN			

TOUCH_REALT	Touch with Real-T function	0:disable & clear real-T stop flag	1:enable
TINT_EN	Touch INT Enable		
TIS	Touch I/O Port input select , 00: NA, 01:PortA, 10:PortB, 11:PortC		
SVT	Touch SVT , 0: 1/4*VCC, 1: 1/2*VCC		
THEN	Touch Enable		
PREC	Touch Pre-charge 1.5V		

###### ◆ System Miscellanea register #5

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC5.L	18H	00	R/W	THO	LVDO	CMP2_TIMER2	CMP1_TIMER1	CMP2_INT_INV	CMP1_INT_INV	CMP2_INT_EN	CMP1_INT_EN	System Miscellanea register #5
MISC5.H	18H	00	R/W	InterPLY_FREQ_DIV								

THO	Touch Comparator Output (Read Only)		
-----	-------------------------------------	--	--

##### Virtual Programming IO Port

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	B12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
PA_TOUCH	0EH	00	RW	PA_TOUCH[3:0]				PortA touch enable bit				
PB_TOUCH	0FH	00	RW	PB_TOUCH[5:0]				PortB touch enable bit				
PCI_TOUCH.L	10H	00	RW	PC_TOUCH[7:0]				PortC0~ PortC7 touch enable bit				
PCI_TOUCH.H	10H	00	RW	PI_TOUCH[1:0]	-	-	PC_TOUCH[11:8]				PortC8~ PortC11 touch enable bit PortI0~ PortI1 touch enable bit	

If THEN is set high, PA\_TOUCH, PB\_TOUCH and PCI\_TOUCH registers are used to set the corresponding touch enable bit of PortA, PortB, PortC and PortI.



### CST1608A(C) 16-bit Speech Processor

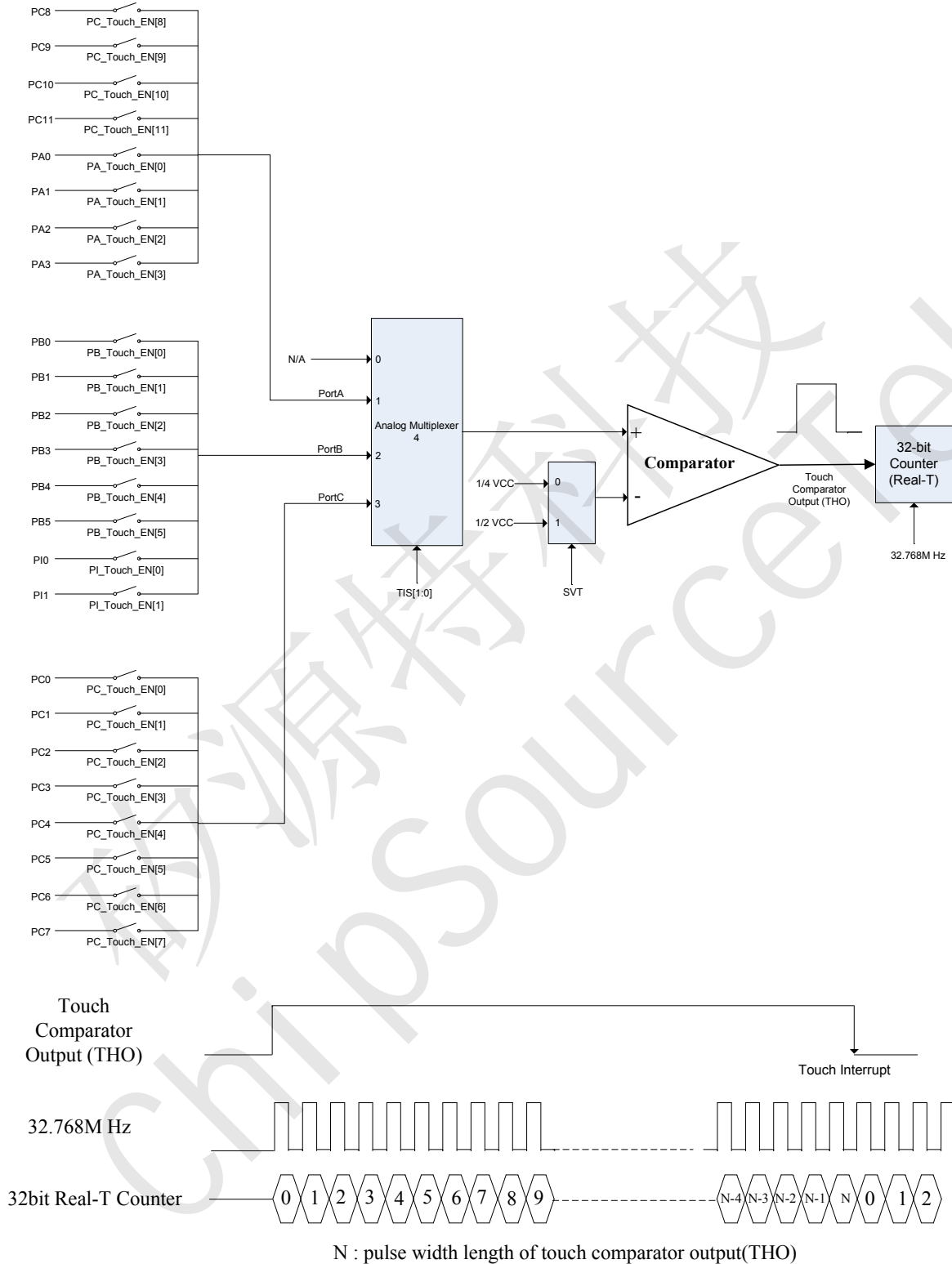


Figure 7.6 Touch Controller Structure



### CST1608A(C) 16-bit Speech Processor

#### 7.4 Audio Output

##### 7.4.1 Mono 16-bit PWM Output

One 16-bit of audio PWM / DAC is built-in CST1608A(C) for mono audio application. Mono output pin PWMP and PWMN are connected to speaker directly. This amplifier can be used to direct drive 8 ohm speaker without any external circuit.

Common I/O registers

Symbol	Adr	Reset	RW	b15/b7	b14/b6	B13/b5	b12/b4	b11/b3	b10/b2	b9/b1	b8/b0	Description
Audio-PWM	16H	XX	W	Audio-PWM[15:0]								16-bit PWM Audio Channel

##### 7.4.2 Mono 16-bit DAC Output

One 16-bit of digital-to-analog converter (DAC) is built-in CST1608A(C) for mono audio application. In order to get more output power driving, which require external amplifier is need.

Common I/O registers

Symbol	Adr	Reset	RW	b15/b7	b14/b6	B13/b5	b12/b4	B11/b3	b10/b2	b9/b1	b8/b0	Description
Audio-DAC	16H	XX	W	Audio-DAC [15:0]								16-bit DAC Audio Channel

##### ◆ System Miscellanea register #4

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC4.L	19H	00	R/W	TOUCH REALT	TINT _EN	TIS	SVT	THEN	PREC	-		System miscellanea register #4
MISC4.H	19H	00	R/W	DAC Bypass	DAC_OG		InterPLY_ EN	Tone_ Mode	InterPLY_FUN			

DAC_OG	Default:0, DAC output gain = 1
--------	--------------------------------

##### 7.4.3 DAC Interpolation Control

One DAC interpolation control unit is built-in CST1608A(C), in order to reduce quantization noise and improve sound quality at DAC audio output.

##### ◆ System Miscellanea register #4

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC4.L	19H	00	R/W	TOUCH REALT	TINT _EN	TIS	SVT	THEN	PREC	-		System miscellanea register #4
MISC4.H	19H	00	R/W	DAC Bypass	DAC_OG		InterPLY_ EN	Tone_ Mode	InterPLY_FUN			

DAC_Bypass	Bypass DAC input function	0:disable(default)	1:enable
InterPLY_EN	0:DAC Interpolation disable	1: DAC Interpolation enable	



### CST1608A(C) 16-bit Speech Processor

Tone_Mode	0:TONE disable 1:TONE enable(Sin wave output)
InterPLY_FUN	0:DAC Interpolation disable 1: DAC Interpolation 8 point 2:DAC Interpolation 16 point 3: DAC Interpolation 32 point

#### ◆System Miscellanea register #5

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	b11/b3	B10/b2	b9/b1	b8/b0	Description
MISC5.L	18H	00	R/W	THO	LVDO	CMP2_TIMER2	CMP1_TIMER1	CMP2_INT_INV	CMP1_INT_INV	CMP2_INT_EN	CMP1_INT_EN	System Miscellanea register #5
MISC5.H	18H	00	R/W	InterPLY_FREQ_DIV								

InterPLY_FREQ_DIV	System Frequency Divisor for DAC interpolation, Default:0
-------------------	---

Setting this value defines how many System Clocks will produce a smooth interpolated value that will be output directly to the DAC. If the system generates a 32-point interpolation operation, and the DAC value is updated every 32 points. Sampling Rate is set by IDE option.

#### Example1:

Sampling Rate = 32k Hz, 16-point interpolation

$$D[7:0](\text{InterPLY\_FREQ\_DIV}) = 65536k \text{ Hz} / (16 \times 32k) = 128(\text{Dec.}) / 0x80(\text{Hex.})$$

#### Example2:

Sampling Rate = 64k Hz, 32-point interpolation

$$D[7:0](\text{InterPLY\_FREQ\_DIV}) = 65536k \text{ Hz} / (32 \times 64k) = 32(\text{Dec.}) / 0x20(\text{Hex.})$$



## CST1608A(C) 16-bit Speech Processor

### 7.5 Auto-FIFO

The Auto-FIFO allows user transfer base on 4-level of data to Audio-PWM. In some case of frame base applications that data transfer is more efficient than sample base. It is advantageous to decrease number of context switch between main program and interrupt service routine (ISR). The FIFO structure reveal as below:

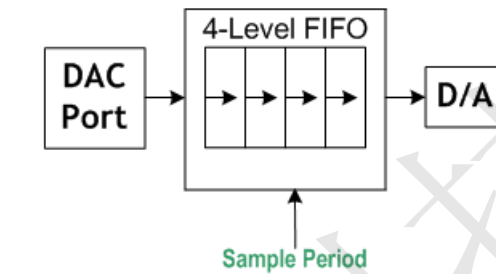


Figure 7.7 Auto FIFO Structure

An interrupt is generated when an entire 4-level FIFO is transfer completed (D/A FIFO buffer is empty), then interrupt service routine should re-load 4-level data to FIFO at ones during 32K or 64K sample period. The FIFO will automatically shift-out data to Audio-PWM at each sample period.

Note: Auto-FIFO is enable/disable by option setup.



## CST1608A(C) 16-bit Speech Processor

### 8. CST1608A(C) Flash Control

#### 8.1 Flash Structure

CST1608A(C) is built in 512Kx16 program/data FLASH memory. This Flash is offered with sector endurance of more than 1,000 cycles, Data retention is rated at greater than 10 years. It is suited for convenient and economical updating of program, configuration, or data memory.

The sector architecture is based on sector size of 256 words/512 bytes. The sector erase operation allows the system to erase the device on a sector. The sector architecture is shown as Figure 8.1.

Sector Select(255 sectors)								Word Select(256 words)							
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Figure 8.1 The sector architecture

#### 8.2 Flash Sector Erase

The Sector Erase instruction needs to assign erasing sector number to AR register. When “PE = AR” instruction is executed, select of sector will be erased.

Example: Erase 16th sector (0x1000 – 0x10FF)

```
AR = 0x0010 // assign erasing sector number
PE = AR     // sector erasing
```

Note: Erasing time of one sector is 2.6ms.

#### 8.3 Flash Word Programming

CST1608A(C) provide one word programming instruction. The Word Programming instruction needs to assign programming data to AR register and assign programming address to P0/P1 register. When “PM[P0/P1] = AR” instruction is executed, select of flash address will be programmed.

Example: Program 16th word address (0x0010)

```
P0.hh = 0x0000
P0 = 0x0010 // assign programming address
AR = 0x5678 // assign programming data
PM[P0] = AR // word programming
```

Note: Programming time of one word is 41us.



## CST1608A(C) 16-bit Speech Processor

### 8.4 Flash Word Read

CST1608A(C) provide word read instruction. The Word Read instruction needs to assign reading address to P0/P1 register. When “AR = PM[P0/P1]” instruction is executed, select of word will be read out to AR register.

Example: Read 16th word address (0x0010)

P0.hh = 0x0000

P0 = 0x0010 // assign reading address

AR = PM[P0] // read word data

**Note: Reading time of one word is two cycles of System Clock.**

### 8.5 Flash unlock/lock for Programming/Erase instruction

Flash unlock instruction need to be executed before flash programming/erase instruction will be executed. This flash unlock instruction is to ensure the safety of the flash operation in order to prevent flash data from being modified by mistake. After the Flash program/erase instruction is executed, the Flash lock instruction needs to be executed to ensure the flash security.

Example1: Flash unlock

ENC\_DAT EQU 54H

ar = 0xdb3f

io[ENC\_DAT] = ar

ar = 0x8a17

io[ENC\_DAT] = ar

Example2: Flash lock

ENC\_DAT EQU 54H

ar = 0x1234

io[ENC\_DAT] = ar

ar = 0x5678

io[ENC\_DAT] = ar



## CST1608A(C) 16-bit Speech Processor

### 9. CST1608A(C) Others

#### 9.1 Dynamic System Clock

Operation frequency can be adjusted by software dynamically. User can adjust operation frequency in order to reduce power consumption.

Write Data	CPU Operation frequency
2	4 MHz
3	6 MHz
4	8 MHz

Example: Change CPU Operation frequency to 4MHz.

```
SPEED EQU 67H
```

```
ar = 0x0002;  
io[SPEED] = ar;
```

#### 9.2 Low Voltage Detector

The voltage detect range of LVD is 1.8V ~ 2.98V for D2ENB = 0 or 2.7V~4.47V for D2ENB=1. When  $VCC/2$  or  $VCC/3 < VBGI$ , LVDO=1, otherwise LVDO=0.

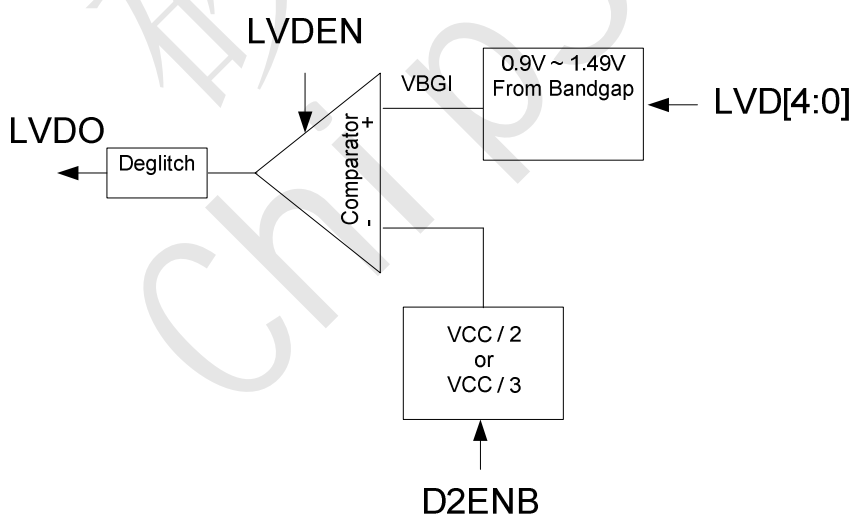


Fig 9.2-1 Function diagram of LVD



### CST1608A(C) 16-bit Speech Processor

Register	MISC3.b6	MISC3.b7	MISC3.b8~12	MISC5.b6
Control signal	LVDEN	D2ENB	LVD[4:0]	LVDO

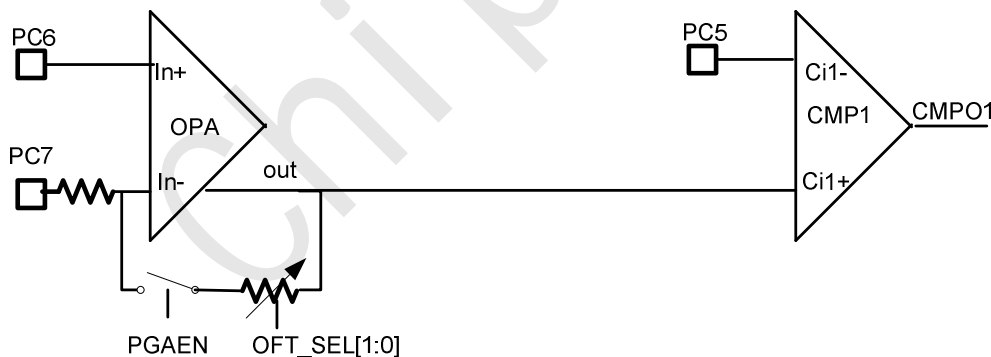
LVD [4:0]	D2ENB=0		D2ENB=1		LVD [4:0]	D2ENB=0		D2ENB=1	
	Ideal	Real	Ideal	Real		Ideal	Real	Ideal	Real
0x00	1.80	<b>1.79</b>	2.70	<b>2.68</b>	0x10	2.44	<b>2.40</b>	3.66	<b>3.59</b>
0x01	1.84	<b>1.83</b>	2.76	<b>2.74</b>	0x11	2.48	<b>2.44</b>	3.72	<b>3.65</b>
0x02	1.88	<b>1.87</b>	2.82	<b>2.79</b>	0x12	2.52	<b>2.48</b>	3.78	<b>3.70</b>
0x03	1.92	<b>1.91</b>	2.88	<b>2.85</b>	0x13	2.56	<b>2.52</b>	3.84	<b>3.76</b>
0x04	1.96	<b>1.95</b>	2.94	<b>2.91</b>	0x14	2.60	<b>2.56</b>	3.90	<b>3.82</b>
0x05	2.00	<b>1.98</b>	3.00	<b>2.96</b>	0x15	2.64	<b>2.59</b>	3.96	<b>3.87</b>
0x06	2.04	<b>2.02</b>	3.06	<b>3.02</b>	0x16	2.68	<b>2.63</b>	4.02	<b>3.93</b>
0x07	2.08	<b>2.06</b>	3.12	<b>3.08</b>	0x17	2.72	<b>2.67</b>	4.08	<b>3.99</b>
0x08	2.12	<b>2.10</b>	3.18	<b>3.13</b>	0x18	2.76	<b>2.71</b>	4.14	<b>4.05</b>
0x09	2.16	<b>2.14</b>	3.24	<b>3.19</b>	0x19	2.80	<b>2.75</b>	4.20	<b>4.10</b>
0x0A	2.20	<b>2.18</b>	3.30	<b>3.25</b>	0x1A	2.84	<b>2.79</b>	4.26	<b>4.16</b>
0x0B	2.24	<b>2.21</b>	3.36	<b>3.30</b>	0x1B	2.88	<b>2.82</b>	4.32	<b>4.22</b>
0x0C	2.28	<b>2.25</b>	3.42	<b>3.36</b>	0x1C	2.92	<b>2.86</b>	4.38	<b>4.27</b>
0x0D	2.32	<b>2.29</b>	3.48	<b>3.42</b>	0x1D	2.96	<b>2.90</b>	4.44	<b>4.33</b>
0x0E	2.36	<b>2.33</b>	3.54	<b>3.48</b>	0x1E	3.00	<b>2.94</b>	4.50	<b>4.39</b>
0x0F	2.40	<b>2.37</b>	3.60	<b>3.53</b>	0x1F	3.04	<b>2.98</b>	4.56	<b>4.44</b>

It's a COB test result for reference only.

### 9.3 2 channel comparators and a two stage OPA with class AB output

Register	MISC2.b10	MISC2.b13	MISC2.b6	MISC2.b7	MISC3.b14/b15
Control signal	OPEN	PGAEN	CMP1EN	CMP2EN	CMPO1/2
One-PGA_One-CMP	0	1	1	0	
One-OP_One-CMP	1	0	1	0	
Two CMPs	0	0	1	1	

\*\*One PGA one comparator mode



Register	Control signal				
MISC2.b4	OFT_SEL0	0	1	0	1
MISC2.b5	OFT_SEL1	0	0	1	1
Gain (V/V)		50	100	150	200

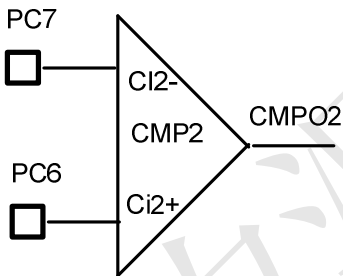
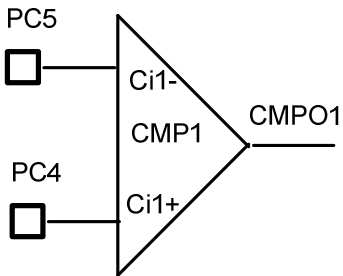


## CST1608A(C) 16-bit Speech Processor

\*\*One OPA one comparator mode



\*\*Two comparators mode



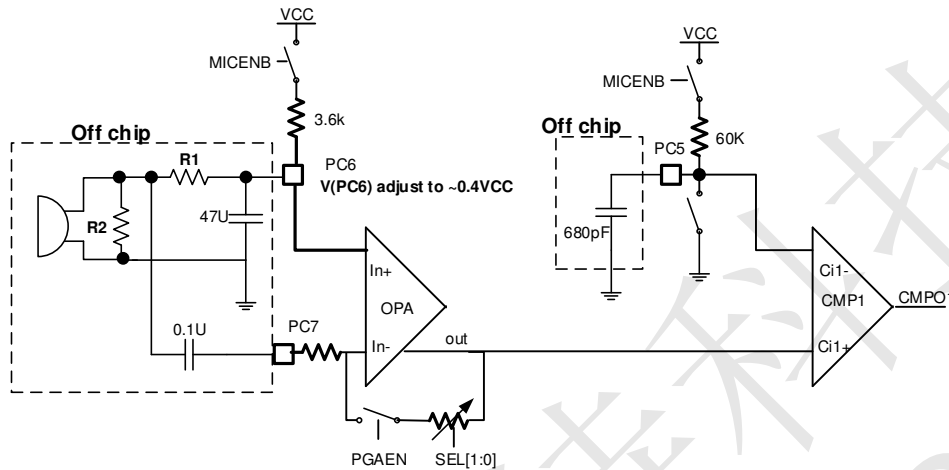


### CST1608A(C) 16-bit Speech Processor

#### 9.4 Microphone with AGC(Auto Gain Control)

Register	MISC2.b10	MISC2.b13	MISC2.b11	MISC2.b6
Control signal	OPEN	PGAEN	RECEN	CMP1EN
Mic-pre-amp_PGA	0	1	1	1
Mic-pre-amp_OP	1	0	1	1

\*\*Mic-preamplifier PGA (support AGC)

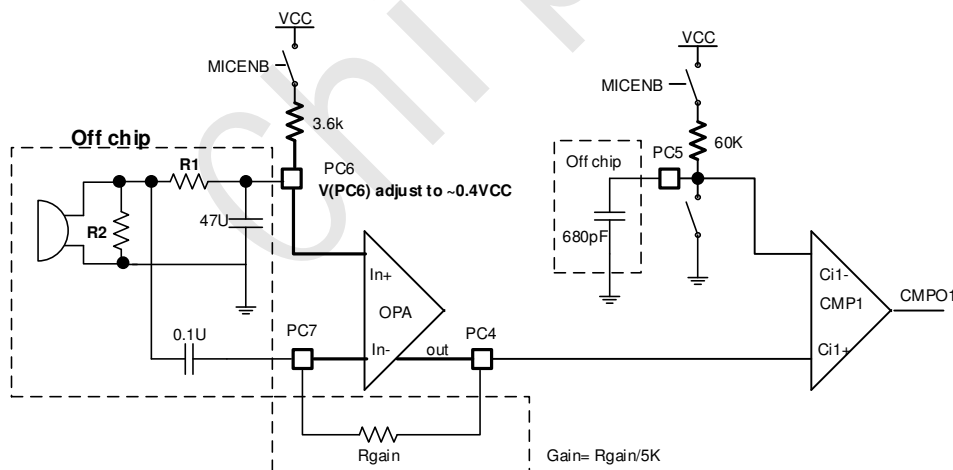


Register	Control signal				
MISC2.b4	OFT_SEL0	0	1	0	1
MISC2.b5	OFT_SEL1	0	0	1	1
Gain (V/V)		50	100	150	200

Recommend value for R1/R2

VCC	R1	R2
2 cell battery	2K	N/A
3 cell battery	2K	7.5K

\*\*Mic-preamplifier OP (no support AGC)

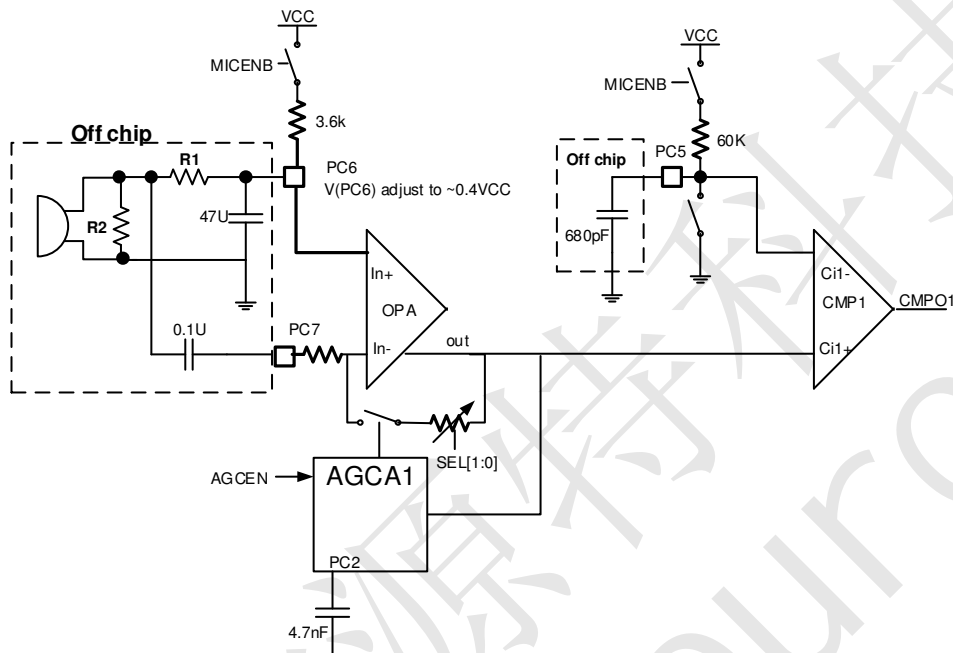




## CST1608A(C) 16-bit Speech Processor

\*\*AGC function (CAGC=4.7nF)

Register	MISC2.b10	MISC2.b13	MISC2.b11	MISC2.b6	MISC2.b14
Control signal	OPEN	PGAEN	RECEN	CMP1EN	BUZEN
Mic-pre-amp_PGA with AGC	0	1	1	1	1
Mic-pre-amp_PGA without AGC	0	1	1	1	0



AGC function can work on PGA mode only.



## CST1608A(C) 16-bit Speech Processor

### 10. CST1608A(C) System Control

#### 10.1 Halt Mode & Wake up

CST1608A(C) is changed into HALT mode (system clock stop, RTC stop by Option setup) when HALT instruction executed. It provides a power saving mode for those applications requiring a very low stand-by current. The RTC timer, PA0~PA5, PB0~PB5 and PC0~PC11 are supporting the wake-up MCU function when related I/O port raising/falling edge which selects by program. The program counter will be 04H when HALT instruction executed immediately; in addition, when wake up condition is occurred, MCU will release HALT state and program counter go-to next address after difference stable clock is delayed by option. During the Halt mode period, the SRAM will keep their previous data without changing.

#### 10.2 Watch Dog Timer Reset (WDT)

Symbol	Adr	Reset	RW	b15/b7	b14/b6	b13/b5	b12/b4	B11/b3	b10/b2	b9/b1	b8/b0	Description
ClrWDT	1DH	XX	W									Clear WDT

The watch dog timer (WDT) is used to reset whole chip when detect unexpected execution sequence of instructions caused by accident condition, avoiding dead lock of MCU program. Software shall run an "clear watch dog timer"( CLRWDT ) instruction before this timer time out. It will generate a reset signal to reset whole system when WDT overflow.

WDT will be reset when wake-up from halt or after power on or software clear. In test mode, watch dog timer will be disabled no matter watch-dog-timer is time-out or not.

The reset watch dog timer code syntax is strongly recommended as: "**CLRWDT = AR**".

#### 10.3 Low Voltage Reset

When VCC power is applied to the chip, the low voltage reset is initially enabled by default, it will be disabled when in halt mode. The internal system reset will be generated if VCC power below the voltage of LVR(option setup). The normal operation of LVR is always enable expect disable in HALT mode.

#### 10.4 Reset System

CST1608A(C) reset is come from four signals which are power on reset, low voltage reset(LVR), external IO PB3 or PC3 pin (by option) and WDT overflow reset, as shown in Figure 10.1. A dedicated external resetb input pin (by option) is provided to reset this chip, which has 30K ohm pull up resistor. The operation frequency of MCU will go back to BANK0 mode when reset occurred.



## CST1608A(C) 16-bit Speech Processor

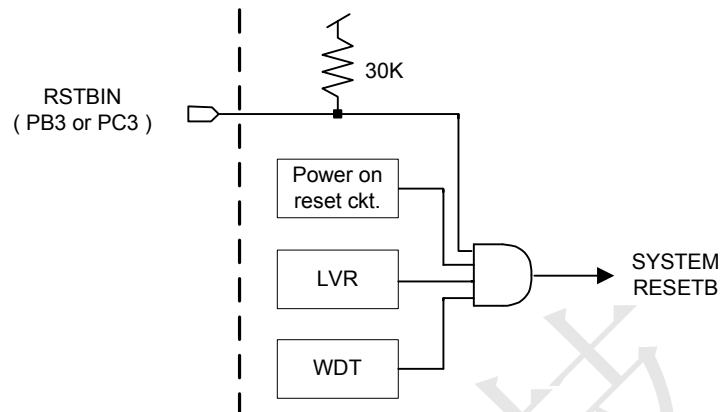


Figure 10.1 Reset system block diagram

### 10.5 Clock System Architecture

CST1608A(C) clock system supports internal ROSC(65.536MHz) for System Clock, and Low power RC oscillator(32768Hz  $-30\%@1.8V \sim +30\%@5.5V$ ) for RTC function.

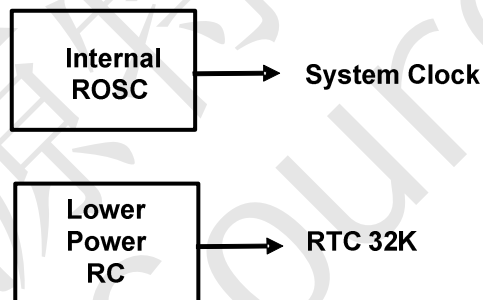


Figure 10.2 Clock System Diagram



### CST1608A(C) 16-bit Speech Processor

#### 11. CST1608A(C) Electrical Characteristics

##### 11.1 Absolute Maximum Rating

DC Supply Voltage	VCC	<5.5	V
Input Voltage	Vin	-0.5 to VCC+0.5	V
Operating Temperature Range	Ta	-20 to 75	
Storage Temperature Range	Tstg	-50 to 150	

##### 11.2 DC/AC Characteristics

VCC=3.0V, Ta=25°C unless otherwise noted

Operating voltage	VCC	1.8 V	-	5.5 V	
Operating frequency (BANK0)	Fbank0	4.096MHz ±3%		8.192MHz ±3%	
Operating frequency (BANK15)	Fbank15		32.768MHz ±3%		
RC oscillator frequency	Frc1		65.536MHz ±3%		
Low power RC oscillator frequency	Frc2	32768Hz -30%		32768Hz +30%	
Normal Sleep Halt Current	lhalt1		5uA	9uA	All function off
Operating Current	Iop		5mA		no load
input high voltage (PortA, PortB, PortC)	VIH	0.6 VCC			Without schmitt trigger
input low voltage (PortA, PortB, PortC)	VIL			0.5 VCC	Without schmitt trigger
input high voltage (PortI)	VIH	0.7 VCC			With schmitt trigger
input low voltage (PortI)	VIL			0.3 VCC	With schmitt trigger
output high voltage	Voh	0.95 VCC			no load
output low voltage	Vol			0.05 V	no load
output high current	Ioh		16 mA		Vout=VCC-0.4V, PortA, B, C, I select strength driving option
output low current	Iol		-16 mA		Vout=0.4V PortA, B, C, I select strength driving option
pull-down resistance	Rpd		50K/220K/ 1M ohm		pins with pull-down, Port A,B,C, I
PWM driving current			350 mA		VPD=5.0V
PWM sink current			400 mA		VPD=5.0V



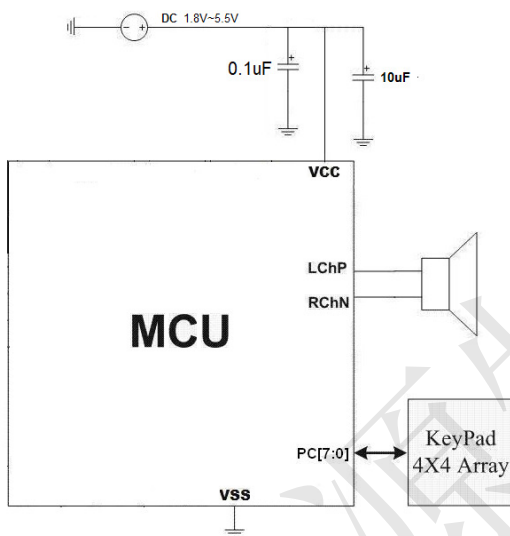
## CST1608A(C) 16-bit Speech Processor

### 12. CST1608A(C) Application Circuit

Application Circuit: Mono16-bit PWM output

**Notice:**

1. **VCC Decoupling Cap 10uF should be close to IC within 0.5cm.**
2. **VCC Decoupling Cap 0.1uF should be close to IC within 0.5cm.**
3. **The width of VCC and VSS power line should be greater than 30 mil in PCB layout.**





## CST1608A(C) 16-bit Speech Processor

### 13. CST1608A(C) Appendix:

#### Appendix 1: PORT A PIN MAP RELATE TO FUNCTION

	PORT A[5:0]					
	5	4	3	2	1	0
2 Ch Comparator						
SPI (Master)						
ICE						
IR 38K			IR_38KO			
EXT-RESETB						
EXT-INT				EXT_INT1	EXT_INT0	

#### Appendix 2: PORT B PIN MAP RELATE TO FUNCTION

	PORT B[5:0]					
	5	4	3	2	1	0
2 Ch Comparator						
SPI (Master)	CS			SO(in)	SI(out)	CLK
ICE						
IR 38K						
EXT-RESETB			EXT-RSTB(0)			
EXT-INT						



### CST1608A(C) 16-bit Speech Processor

#### Appendix 3: PORT C PIN MAP RELATE TO FUNCTION

	PORT C[7:0]							
	7	6	5	4	3	2	1	0
2 Ch Comparator	CAP2-	CAP2+	CAP1-	CAP1+		AGC		
SPI (Master)								
ICE								
IR 38K								
EXT-RESETB					EXT-RSTB(I)			
EXT-INT								

	PORT C[11:8]			
	11	10	9	8
2 Ch Comparator				
SPI (Master)				
ICE				
IR 38K				
EXT-RESETB				
EXT-INT				

#### Appendix 4: PORT I PIN MAP RELATE TO FUNCTION

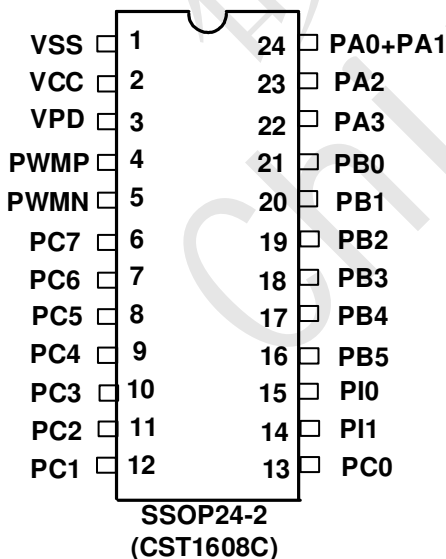
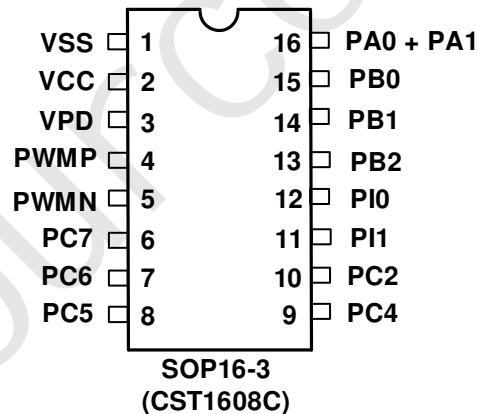
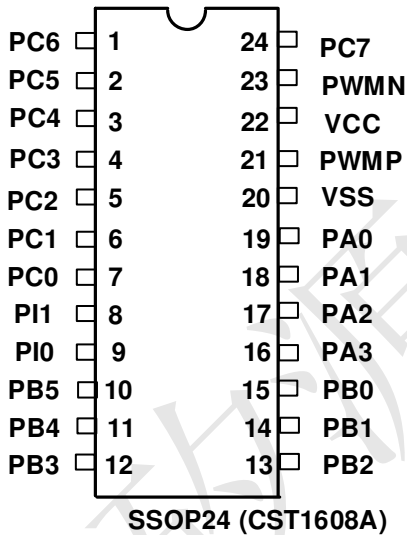
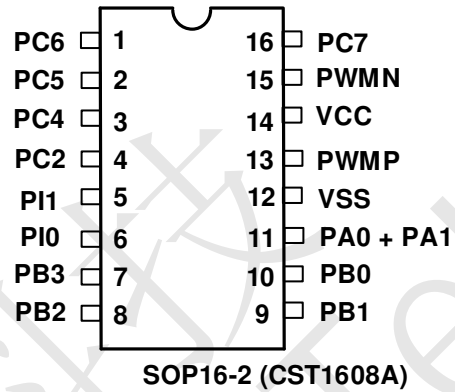
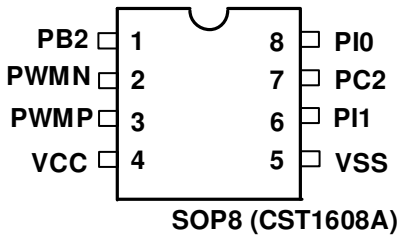
	PORT I[1:0]	
	1	0
2 Ch Comparator		
SPI (Master)		
ICE	ICE_SCLK(I)	ICE_SD(I/O)
IR 38K		
EXT-RESETB		
EXT-INT		



CST1608A(C) 16-bit Speech Processor

14. CST1608A(C) Package:

SOP8 / SOP16-2 / SOP16-3 / SSOP24 / SSOP24-2

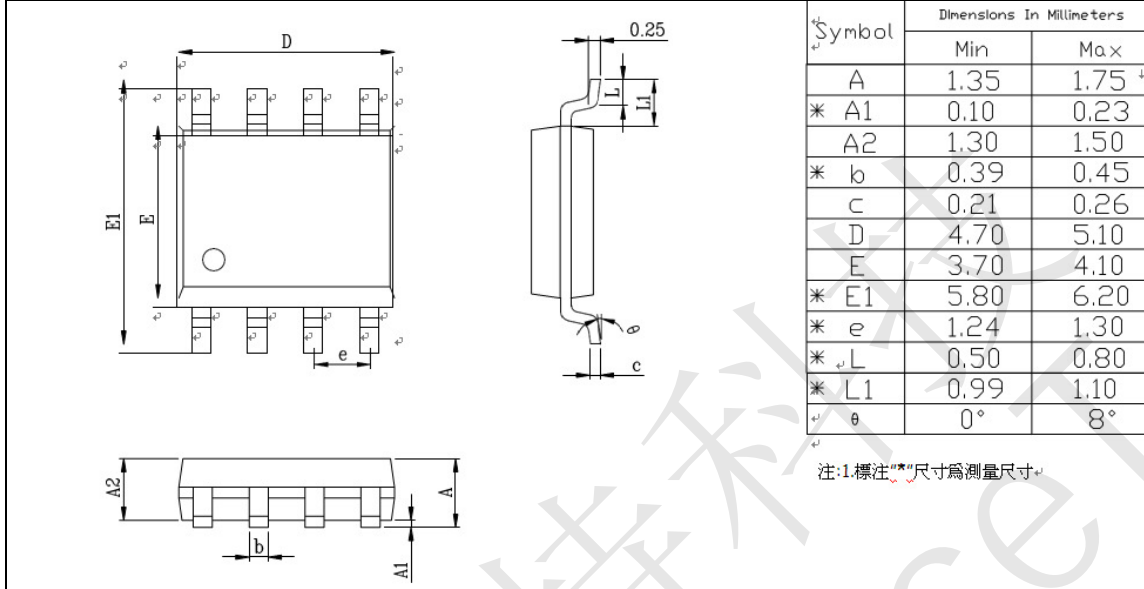




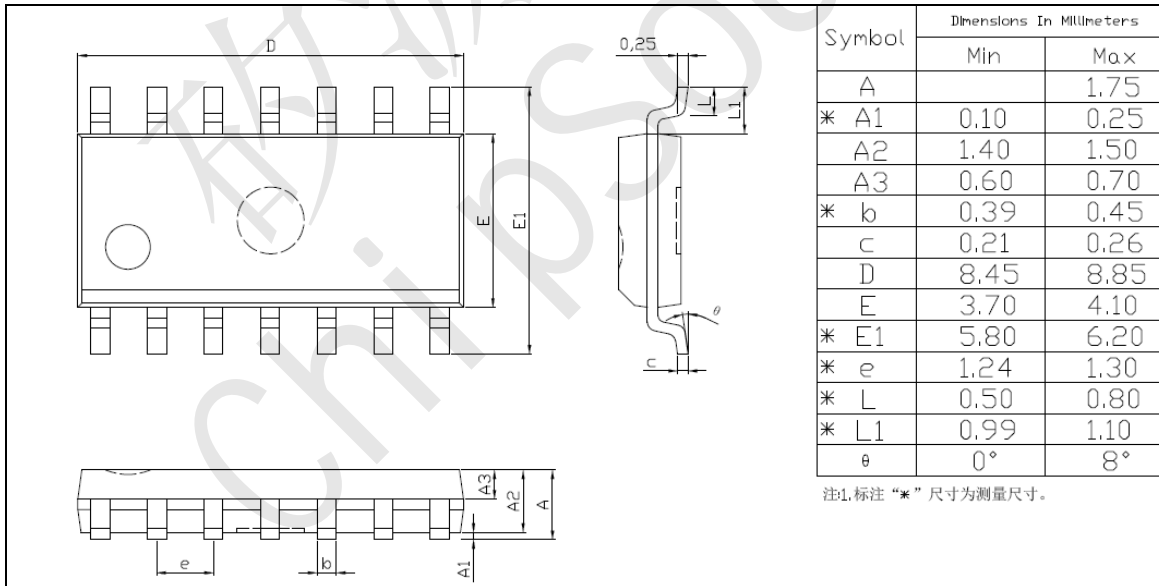
CST1608A(C) 16-bit Speech Processor

PACKAGE INFORMATION

SOP08



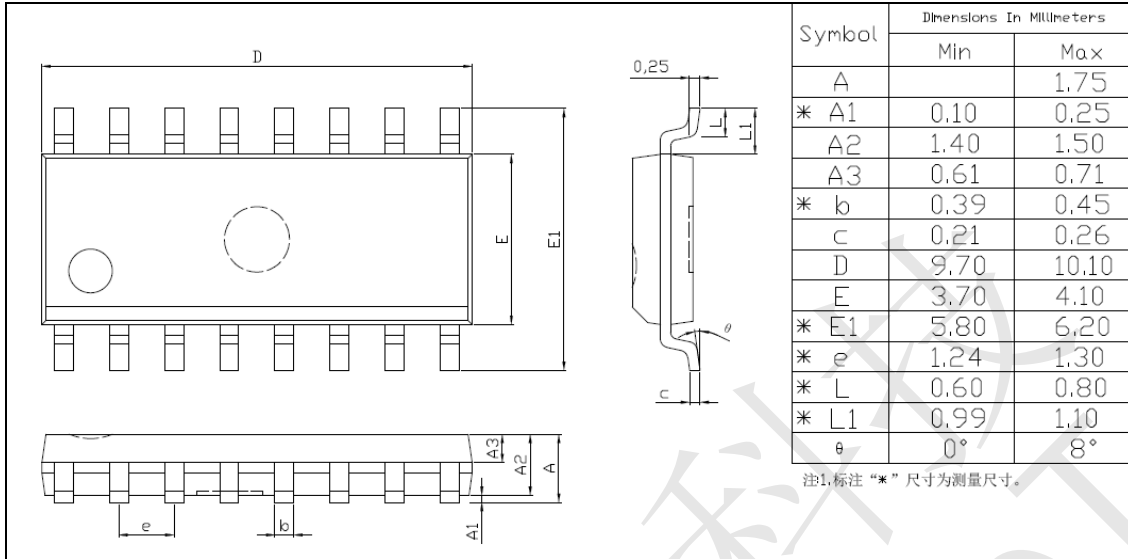
SOP14



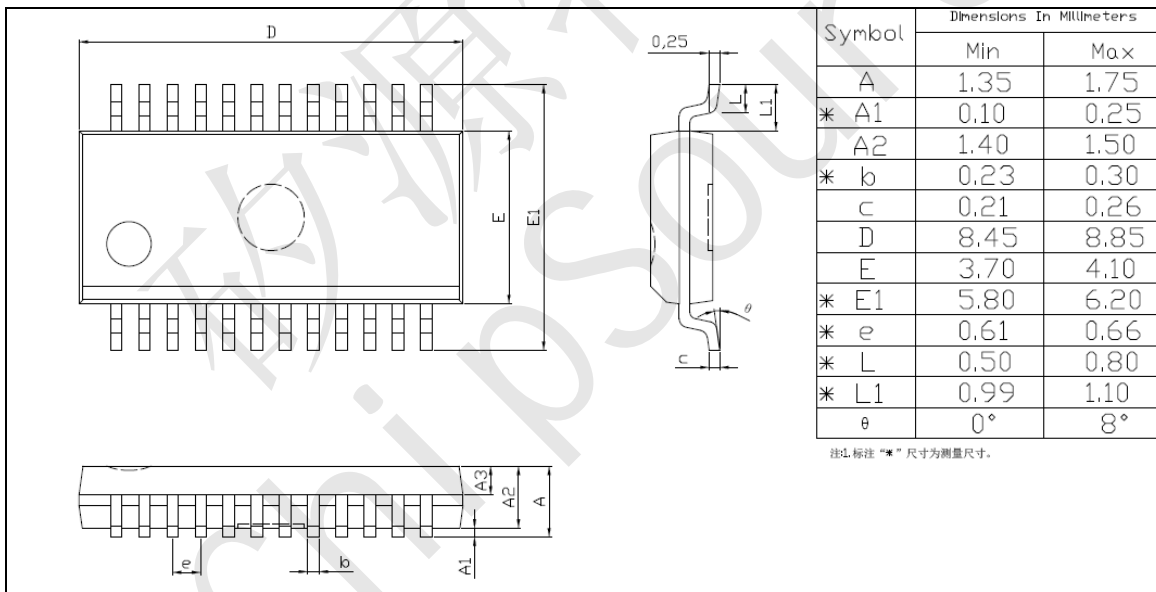


CST1608A(C) 16-bit Speech Processor

SOP16



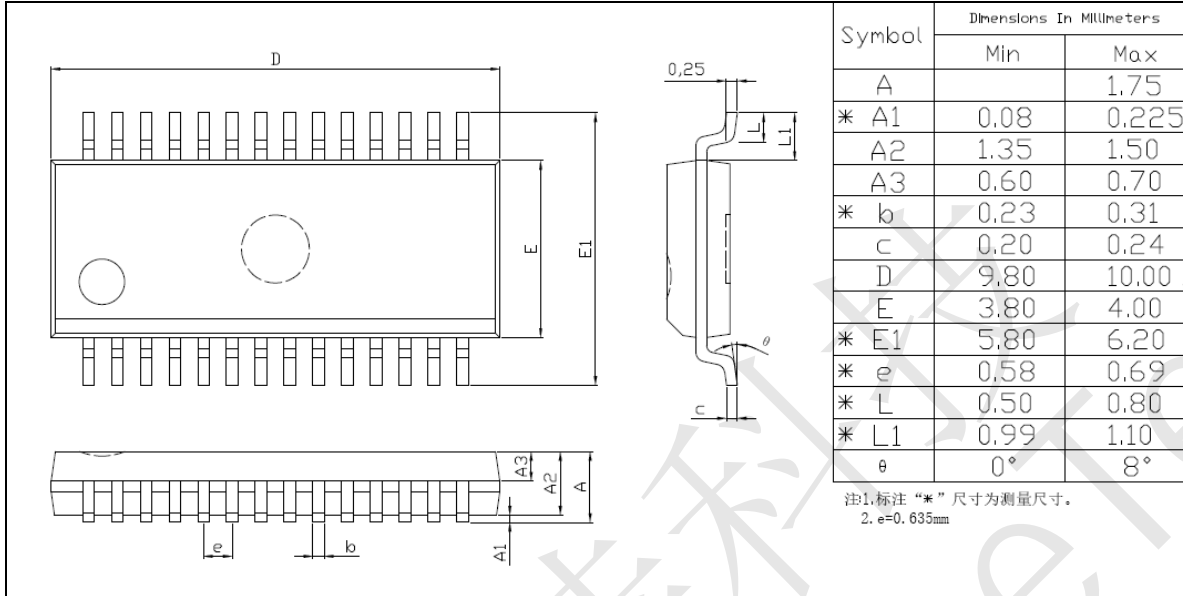
SSOP24





CST1608A(C) 16-bit Speech Processor

SSOP28





## CST1608A(C) 16-bit Speech Processor

### 15. CST1608A(C) SPI Operation Sequence:

#### A1.0 Initial Process

```
set io[IOC_PB].b5      // set output port
set io[STATUS].b8     // Enable SPI Control

set io[PortB].b5      // CS=1
ar = io[SPI_CTRL]
set ar.b11             // set ICS = 1
io[SPI_CTRL] = ar
```

#### A1.1 Sending Data Process

```
clr io[PortB].b5      // CS = 0
ar = io[SPI_CTRL]
clr ar.b11            // ICS = 0
io[SPI_CTRL] = ar
```

#### SPI\_write\_data:

```
ar = data
io[SPI_DATA] = ar;    // write data

ar = 0x12;             // set total byte number & send data
ah = 0x10;             // speed 16M
io[SPI_CTRL] = ar;    //set SEND = 1;
call Check_Tran_OK

set io[PortB].b5      // CS = 1
ar = io[SPI_CTRL]
set ar.b11            // ICS = 1
io[SPI_CTRL] = ar
```

#### Check\_Tran\_OK:

```
ar = io[SPI_CTRL]
test ar.b7
if eq jmp Check_Tran_OK
```

#### Check\_Tran\_OK\_End:

```
rets
```



## CST1608A(C) 16-bit Speech Processor

### A1.2 Receiving Data Process

```
clr io[PortB].b5           // CS =0
ar = io[SPI_CTRL]
clr ar.b11                 // ICS = 0
io[SPI_CTRL] = ar
```

### SPI\_read\_data:

```
ar = 0x22;                // set total byte number & receive data
ah = 0x10;                // speed 16M
io[SPI_CTRL] = ar;        //set RCV = 1;
call Check_Tran_OK
ar = io[SPI_DATA];        // read data

set io[PortB].b5          // CS =1
ar = io[SPI_CTRL]
set ar.b11                // ICS = 1
io[SPI_CTRL] = ar
```

### Check\_Tran\_OK:

```
ar = io[SPI_CTRL]
test ar.b7
if eq jmp Check_Tran_OK
```

### Check\_Tran\_OK\_End:

```
rets
```



## CST1608A(C) 16-bit Speech Processor

### 16. CST1608A(C) Revision history

REVISION	DESCRIPTION	PAGE	DATE
V1.0	New establish		2022/5/31
V1.1	PWM / DAC are forbidden to be connected to 4Ω speaker.	4	2022/6/27
V1.2	CST1608A(C) do not support ultra-deep sleep.	3	2022/11/8
	PortC2 are forbidden to be connected with pull-up resistance.	3, 6	
	Remove SOP16 package	47	
	New add SOP16-3 / SSOP24-2 package For CST1608C	47	
V1.3	Remove Buzzer and speaker wake up. (MISC1 bit10, MISC2 bit 12 and MISC2 bit14 should be set to 0. )	9, 10, 11	2022/12/28
V1.4	Add CST1608A comment to package chapter	46	2023/1/16
V1.5	PortC2 are forbidden to be connected with VCC or pull-up resistance.	3, 6	2023/4/17
	Address 1DH: RealT register	9, 11	
	Add MISC2.b14 BUZEN(AGC) control bit	9, 11	
	Modify 9.4 about BUZEN(AGC) control bit	38, 39	
V1.6	Remove ATT[1:0] and THR[1:0] signal of AGCA1 block.	39	2023/7/12