



CST1602B

Data Sheet

V1.0

128K Embedded Flash

Hi-Performance 16-bit Speech Processor



1. General Description	2
2. Features.....	3
3. Application Field.....	4
4. Block Diagram.....	4
4.1 Pin Assignments /Description.....	5
5. Function Descriptions.....	6
5.1 TxP16S2	6
5.2 TxP16S2 Registers	6
5.2.1 Special Registers	7
5.2.2 Common I/O Registers	8
5.2.3 Basic System Registers	9
5.3 PC Stack	12
5.4 Interrupt	12
5.4.1 Interrupt Vector Table.....	12
5.4.2 Interrupt Controller	13
5.4.3 Interrupt Processing.....	14
5.5 MAC (16-bit X 16-bit Multiplier and Accumulator)	15
6. Memory Configuration.....	17
6.1 Internal Program/Parameter Memory.....	17
6.2 Internal Data Working SRAM	17
6.3 Data Stack.....	18
7. Peripherals.....	19
7.1 Programmable Timers.....	19
7.1.1 Audio PWM Timer	19
7.1.2 Timer1 & Timer2	19
7.1.3 RTC(Real Time Clock) Timer	20
7.2 General Purpose I/O Ports.....	21
7.3 Extension Device	23
7.3.1 SPI Master Controller	23
7.3.2 PortA3 for IR 38KHz Modulation.....	25
7.3.3 Touch Controller.....	26
7.4 Audio Output	28
7.4.1 Mono 16-bit PWM Output	28
7.5 Auto-FIFO	29
8. Flash Control.....	30
8.1 Flash Structure.....	30
8.2 Flash Sector Erase	30
8.3 Flash Word Programming	30



8.4 Flash Word Read	31
8.5 Flash unlock/lock for Programming/Erase instruction	31
9. Others	32
9.1 Dynamic System Clock	32
9.2 Low Voltage Detector	33
10. System Control	34
10.1 Halt Mode & Wake up	34
10.2 Watch Dog Timer Reset (WDT)	34
10.3 Low Voltage Reset	34
10.4 Reset System.....	34
10.5 Clock System Architecture	35
10.6 Ultra Deep Sleep.....	36
10.6.1 Control Registers	36
10.6.2 Flow Chart	37
10.6.3 Example Code	38
11. Electrical Characteristics	41
11.1 Absolute Maximum Rating.....	41
11.2 DC/AC Characteristics	41
12. Application Circuit and Chip Power Circuit.....	42
13. Appendix:	43
14. Package: SOP8 / SOP16 / SSOP24 / SOP16-2	45
15. SPI Operation Sequence	49
16. Revision history	51



CST1602B *16-bit Speech Processor*

1. CST1602B General Description

The TxP16S2™ is a high performance 16-bit MCU, running up to 13MHz and provided with 128K FLASH and total 3K 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 CST1602B is equipped with TxP16S2 and integrating input/output ports, Audio PWM, Timer and Low Voltage Reset...etc. on a chip.

Furthermore, CST1602B extend its external device connection capability such as Serial ROM/Flash. The internal memory capacity includes 128Kx16 program/data FLASH plus 3Kx16 working SRAM.

芯源特科技
ChipSourceTek

CST1602B 16-bit Speech Processor

3. CST1602B Application Field

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

4. CST1602B Block Diagram

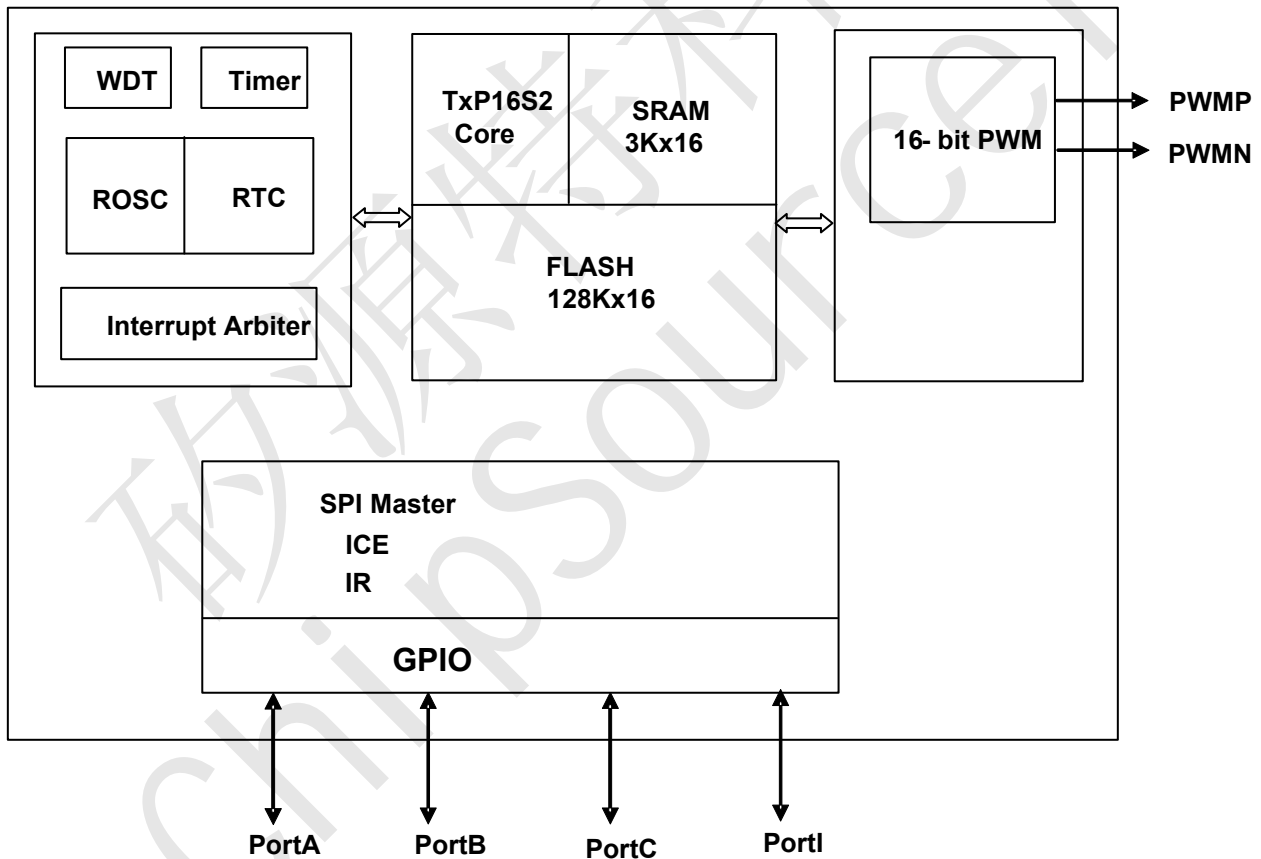


Figure 4.1



CST1602B 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
General Purpose I/O Port			
PortA[3:0]	I/O	Low	PortA is programmable Input/Output port
PortB[5:0]	I/O	Low	PortB is programmable Input/Output port
PortC[7:0]	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(+)
PWMN	O	Low	Digital PWM output(-)

5. CST1602B 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 MAC and circular buffer operations for complexity digital signal processing.

5.2 TxP16S2 Registers

The TxP16S2 contains of register files are illustrated below:

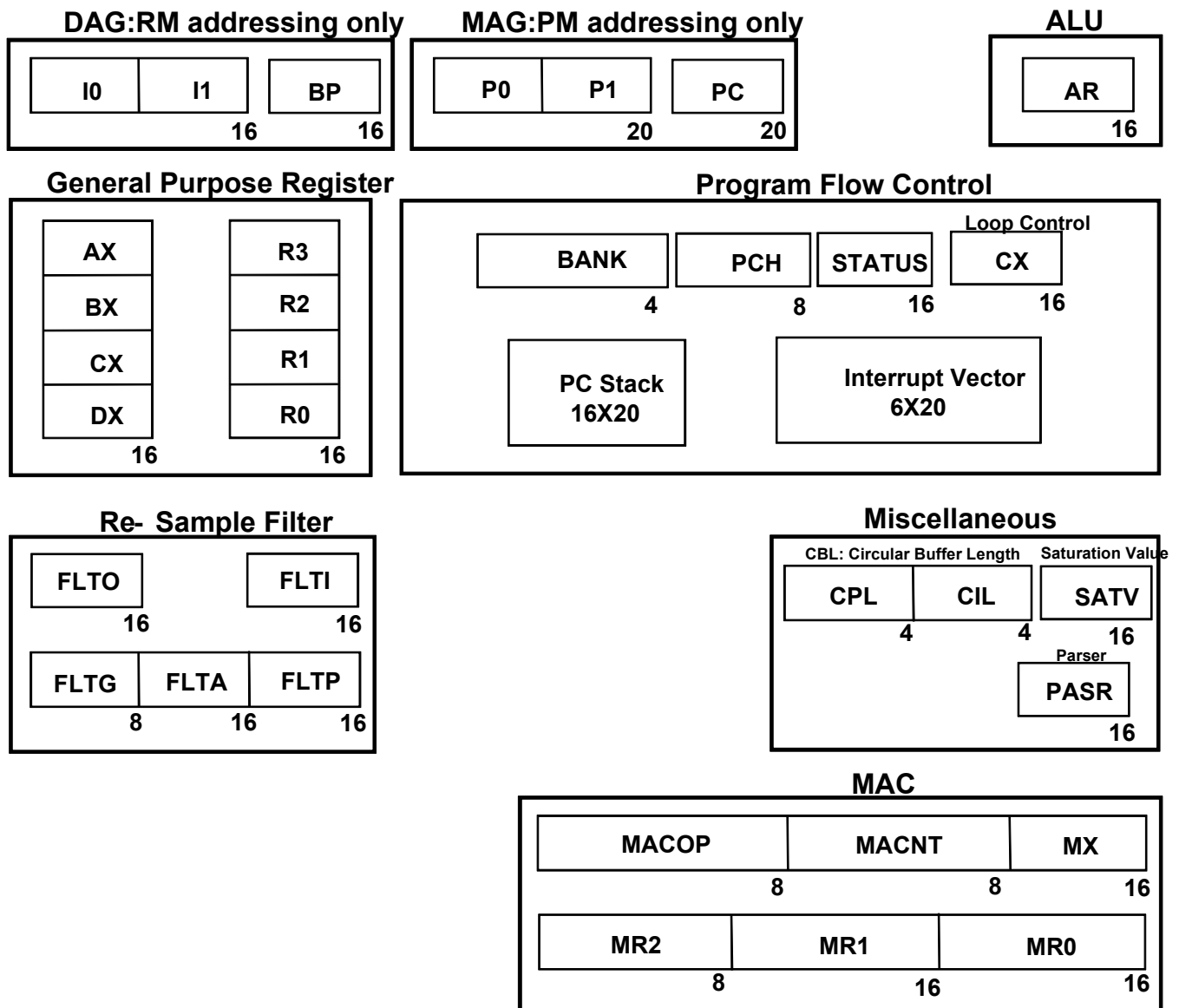


Figure 5.1 TxP16S2 Processor Core Registers

■ REGISTER FILES DEFINE:

AR: Accumulator Register

I0: Index 0 Register

I1: Index 1 Register

BP: Base Pointer Register

P0: Pointer 0 Register

P1: Pointer 1 Register

MACOP: MAC Operation Register

MACNT: MAC Operation Loop Counter

MX: MUL/MAC Input X Register

MR0: MUL/MAC Result Register 0

MR1: MUL/MAC Result Register 1

MR2: MUL/MAC Result Register 2

AX: General AX Register

BX: General BX Register

CX: General CX Register

DX: General DX Register

R0: General R0 Register

R1: General R1 Register

R2: General R2 Register

R3: General R3 Register

CBL: Circular Buffer Length Register

PASR: Parser Register

5.2.1 Special Registers



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	-	-	-	-	-	-	-	-	



CST1602B 16-bit Speech Processor

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.

```
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 #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	-	-	-	-	-	-	System miscellanea register #3
MISC3.H	1AH	00	R/W	-	-	-	LVD				-	

Item	Description
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

◆ 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									

Item	Description
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	-	-	-	-	-	-	System Miscellanea register #5
MISC5.H	18H	00	R/W									

Item	Description
THO	TOUCH Output (Read Only)
LVDO	LVD Output (Read Only)



CST1602B 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
Reserved	04H	-	-	-								
PA_PD50K	05H	00	W	PA_PD50K[3:0]								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	CO	W	PI_PD50K[1:0]	-	-	-	-	-	-	-	PortCI Pull Down 50K Enable <i>(PortI default : Enable PD50K)</i>
PA_PD220K	08H	00	W	PA_PD220K[3:0]								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]	-	-	-	-	-	-	-	PortCI Pull Down 220K Enable
PA_PD1M	0BH	00	W	PA_PD1M[3:0]								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]	-	-	-	-	-	-	-	PortCI Pull Down 1M Enable
PA_TOUCH	0EH	00	W	PA_Touch_EN[3:0]								PortA Touch Enable
PB_TOUCH	0FH	00	W	PB_Touch_EN[5:0]								PortB Touch Enable
PCI_TOUCH.L	10H	00	W	PC_Touch_EN[7:0]								PortCI Touch Enable
PCI_TOUCH.H	10H	00	W	PI_Touch_EN[1:0]	-	-	-	-	-	-	-	PortCI Touch Enable
Reserved	11H	-	-	-								
Reserved	12H	-	-	-								
Reserved	13H	-	-	-								
WAKEN_PA	14H	00	W	PortA_WAKEN[3:0]								PortA WAKE UP Enable
WAKEN_PB	15H	00	W	PortB_WAKEN[5:0]								PortB WAKE UP Enable
WAKEN_PC	16H	00	W	PortC_WAKEN[7:0]								PortC WAKE UP Enable
WAKELV_PA	17H	00	W	PortA_WAKELV[3:0]								PortA WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PA bit
WAKELV_PB	18H	00	W	PortB_WAKELV[5:0]								PortB WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PA bit
WAKELV_PC	19H	00	W	PortC_WAKELV[7:0]								PortC WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PC bit
WAKEDLV_PB	1AH	00	W	PortB_WAKEDLV[3:0]								PortB0 ~ PortB3 Double-Edge WAKE UP Enable
DEEP_SLEEP2.L	1FH	04	R/W	-	-	-	-	-	VDD1EN	CLK	DSENIEN	Deep Sleep2 Status Register[7:0]
DEEP_SLEEP2.H	1FH	00	R/W	DSENO	-	-	-	-	WAKE[3:0]			Deep Sleep2 Status Register[15:8]

5.3 PC Stack

The PC STACK is TxP16S2 special embedded memory used to save (PC+1) value, which is composed with 8-level.

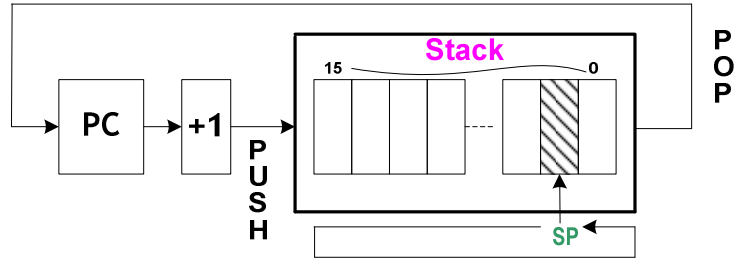


Figure 5.3 PC Stack Structure

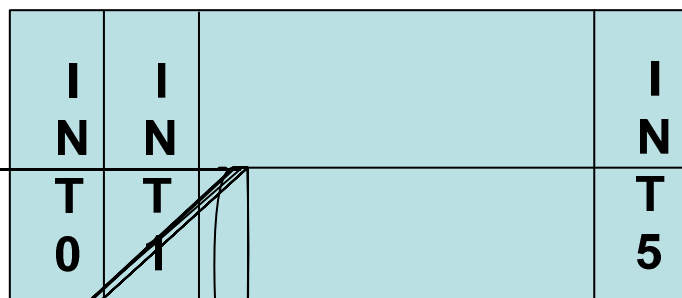
Stack's top value is indexed by stack pointer (SP) register. When CALL instruction is executed, then the (PC+1) will PUSH onto stack addressing by SP and it will auto decrement. At the end of subroutine when RETS instruction is executed the SP will auto increment and stack content of pointer by SP will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

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

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.





CST1602B 16-bit Speech Processor

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 (Note)	1H	INT1_IRQ
Timer2 / ExtINT1 / SPI Master (Note)	2H	INT2_IRQ
RTC Timer / SPI Master (Note)	3H	INT3_IRQ
Reserved	4H	INT4_IRQ
Touch / SPI Master (Note)	5H	INT5_IRQ (lowest)

Table 5.3 Interrupt Sources

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

(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)



CST1602B 16-bit Speech Processor

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.

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.

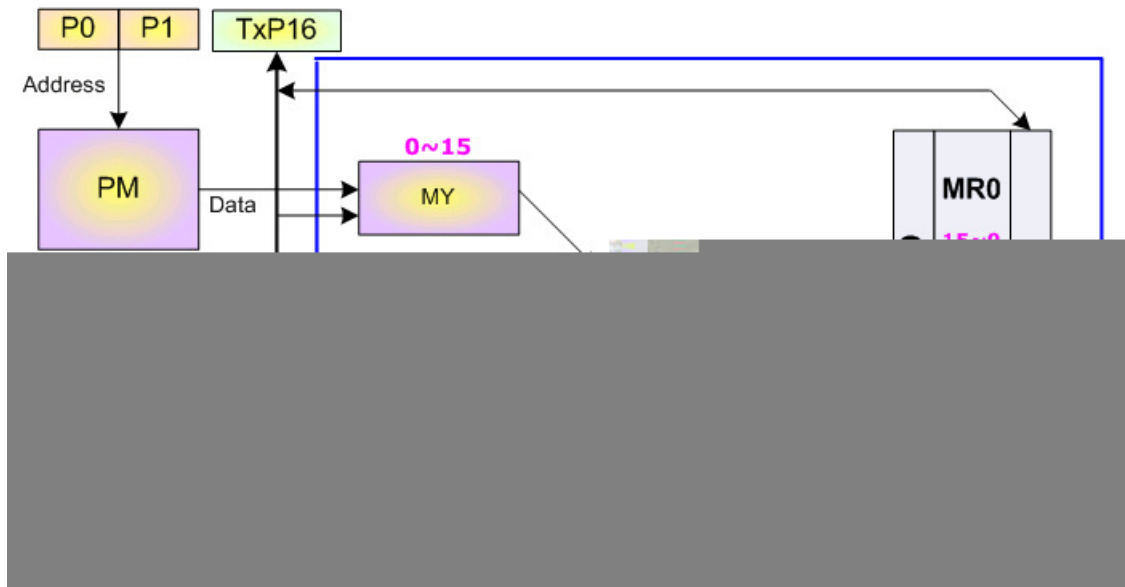


Figure 5.5 MAC Architecture

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)	MX: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:

MY fixed from PM[pointer operation]

Therefore, MAC array operation like this:

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

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

.
.
.

Successive 64 times

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

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)	MX:SU(01)		MAC Operation Setup

CST1602B 16-bit Speech Processor

6. CST1602B 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). The BANK0 and BANK1 are implemented as Flash memory. From BANK2 to BANK15 are system reserved. More details control will describe as follows:

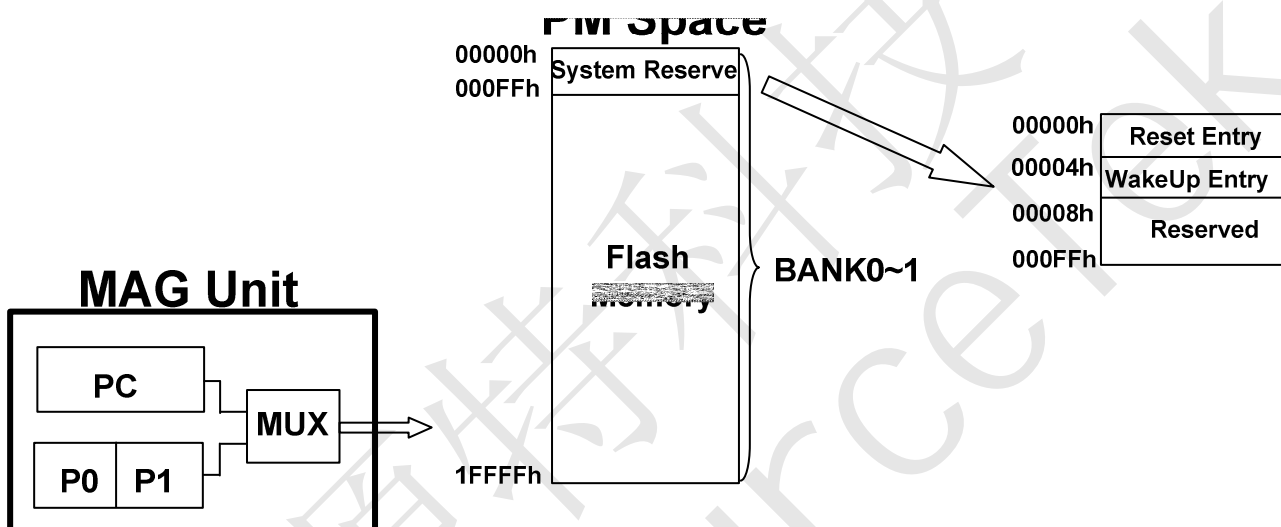
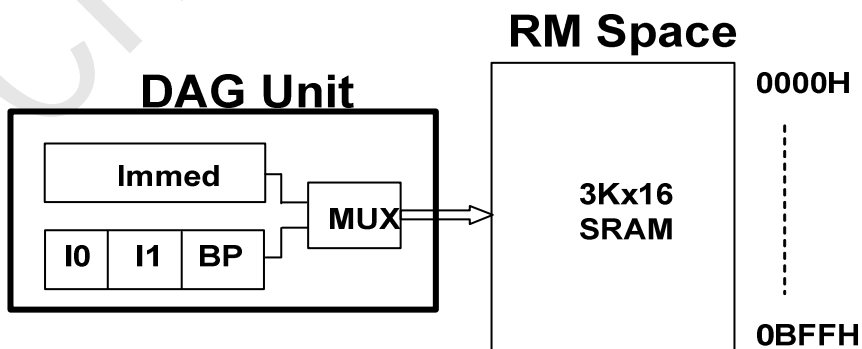


Figure 6.1 PM block diagram

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.

6.2 Internal Data Working SRAM

The internal data working ram space is totally 3Kx16-bit that named as RM. Addressing ranged from 0x0000 through 0x0BFF, 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(0~3071); therefore, RM address will be warping when index value exceed 4095.



CST1602B 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 0x0BFF.

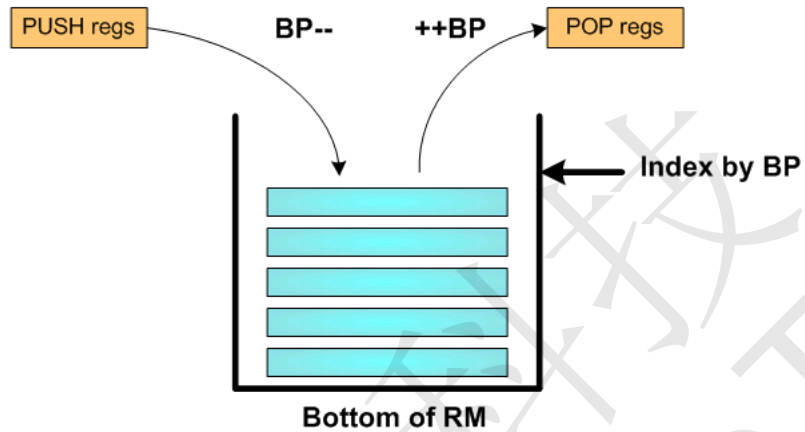


Figure 6.3 Data Stack Structure

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".

7. CST1602B Peripherals

7.1 Programmable Timers

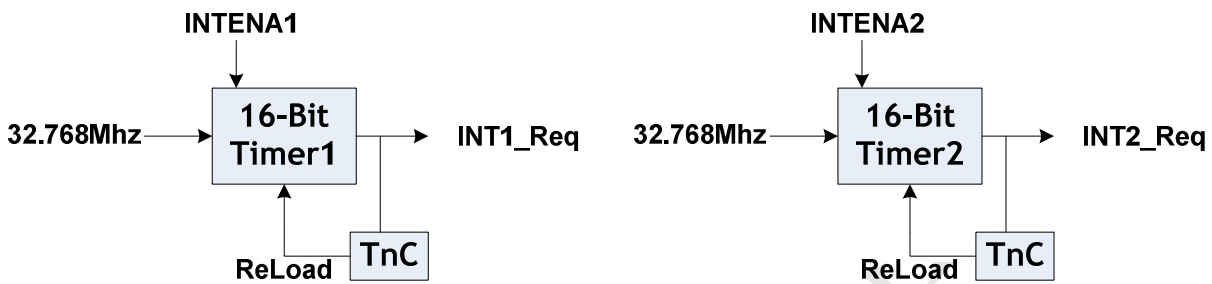
Common I/O registers

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
--------	-----	-------	----	--------	--------	--------	--------	--------	--------

CST1602B 16-bit Speech Processor



$$\text{Int1_Req} / \text{Int2_Req} = (32.768\text{Mhz}) / (\text{TnC}+1)$$

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 -5%@2.0V ~ +15%@5.0V (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)

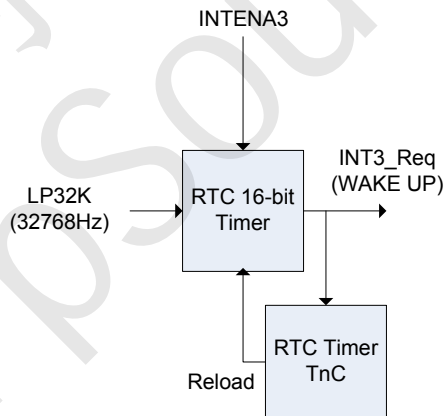


Figure 7.3 RTC Timer Structure



CST1602B 16-bit Speech Processor

7.2 General Purpose I/O Ports

The CST1602B provides 4 I/O ports for user application. There are four I/O port, PA0~PA3, PB0~PB5, PC0~PC7 and PI0~PI1. The input/output bits programmable by IOC control register respectively. PA0~PA3, PB0~PB5, PC0~PC7 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[3:0]						"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]									1" = out, "0" = in of related PI bit
PortA	07H	XX	R/W				PortA[3:0]						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]									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[3:0]						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]		-	-	-	-	-	-	PortCI Pull Down 50K Enable (PortI default : Enable PD50K)	
PA_PD220K	08H	00	W				PA_PD220K[3:0]						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]		-	-	-	-	-	-	PortCI Pull Down 220K Enable	
PA_PD1M	0BH	00	W				PA_PD1M[3:0]						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]		-	-	-	-	-	-	PortCI Pull Down 1M Enable	
PA_TOUCH	0EH	00	W				PA_Touch_EN[3:0]						PortA Touch Enable
PB_TOUCH	0FH	00	W				PB_Touch_EN[5:0]						PortB Touch Enable
PCI_TOUCH.L	10H	00	W				PC_Touch_EN[7:0]						PortCI Touch Enable
PCI_TOUCH.H	10H	00	W	PI_Touch_EN[1:0]		-	-	-	-	-	-	PortCI Touch Enable	
Reserved	11H	-	-										
Reserved	12H	-	-										
Reserved	13H	-	-										
WAKEN_PA	14H	00	W				PortA_WAKEN[3:0]						PortA WAKE UP Enable
WAKEN_PB	15H	00	W				PortB_WAKEN[5:0]						PortB WAKE UP Enable
WAKEN_PC	16H	00	W				PortC_WAKEN[7:0]						PortC WAKE UP Enable
WAKELV_PA	17H	00	W				PortA_WAKELV[3:0]						PortA WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PA bit
WAKELV_PB	18H	00	W				PortB_WAKELV[5:0]						PortB WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PA bit
WAKELV_PC	19H	00	W				PortC_WAKELV[7:0]						PortC WAKE UP Edge "1" = Pos-Edge, "0" = Neg-Edge of related PC bit
WAKEDLV_PB	1AH	00	W				PortB_WAKEDLV[3:0]						PortB0 ~ PortB3 Double-Edge WAKE UP Enable

These totally 20 I/O pins work not only just a general input/output port function but also can be configured as

CST1602B 16-bit Speech Processor

SPI master, Analog input, IR 38K, External interrupt etc. For more detail please refer to relative section.

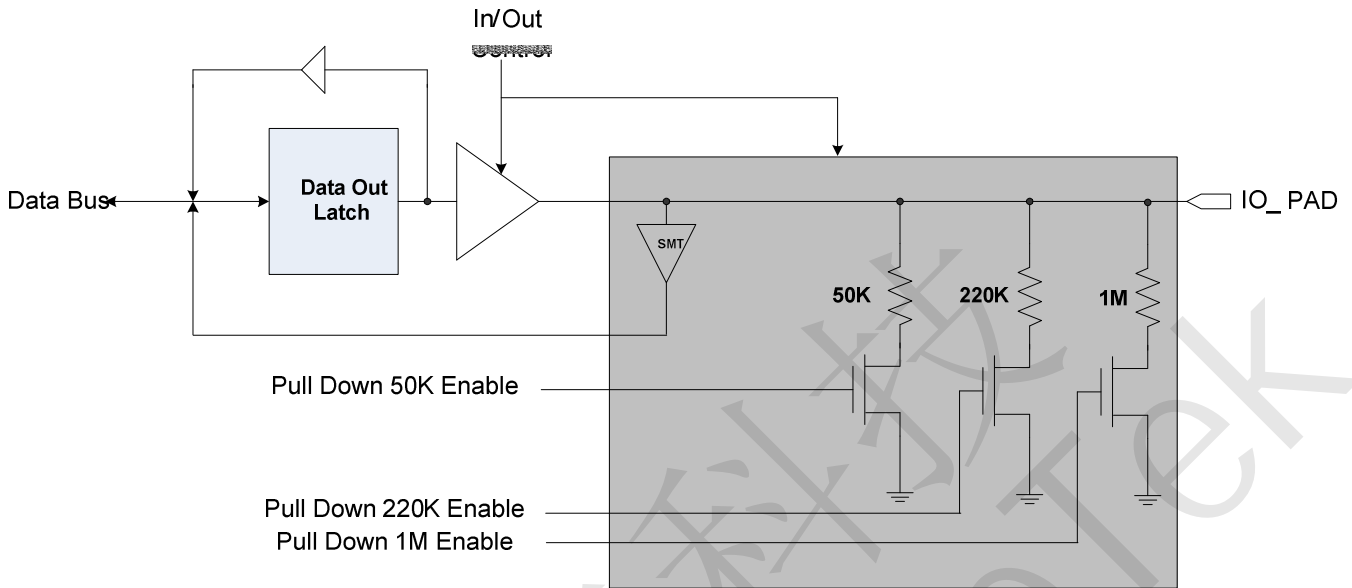


Figure 7.4 Basic I/O Configuration

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]							PortB0 ~ PortB3 Double-WAKE UP 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.

7.3 Extension Device

CST1602B 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.

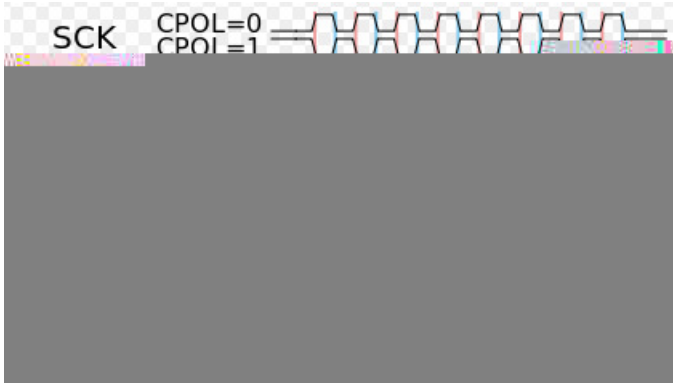
7.3.1.1 Features

- 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

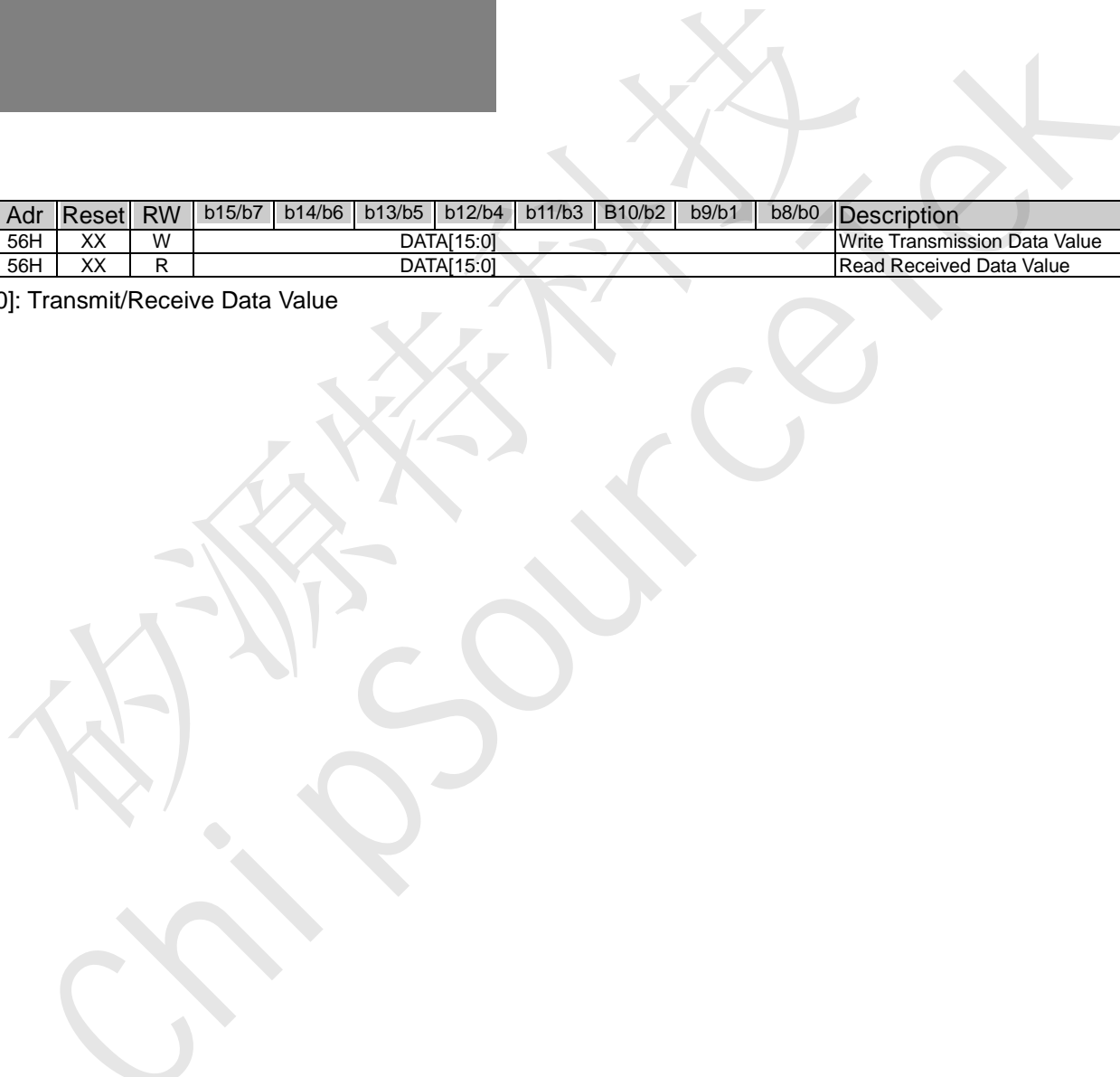


CST1602B 16-bit Speech Processor



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





CST1602B 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.

ChipSourceTek



CST1602B 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.

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									

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	-	-	-	-	-	-	System Miscellanea register #5

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_EN[3:0]				PortA touch enable bit				
PB_TOUCH	0FH	00	RW	PB_Touch_EN[5:0]				PortB touch enable bit				
PCI_TOUCH.L	10H	00	RW	PC_Touch_EN[7:0]				PortC0~ PortC7 touch enable bit				
PCI_TOUCH.H	10H	00	RW	PI_Touch_EN [1:0]	-	-	-				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.

CST1602B 16-bit Speech Processor

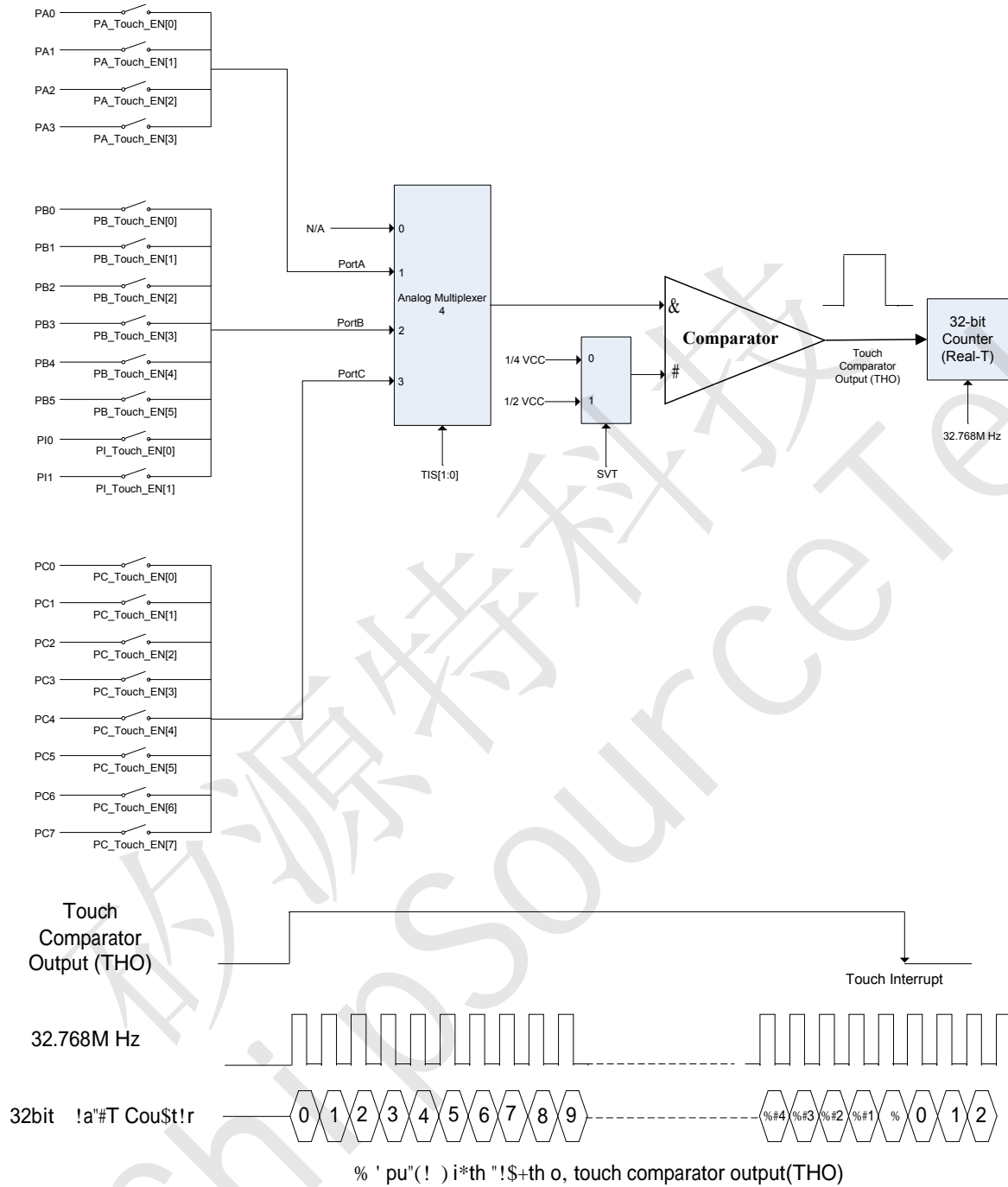


Figure 7.6 Touch Controller Structure



CST1602B 16-bit Speech Processor

7.4 Audio Output

7.4.1 Mono 16-bit PWM Output

One 16-bit of audio PWM is built-in CST1602B 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	PWM[15:0]								16-bit PWM Channel

CST1602B 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:

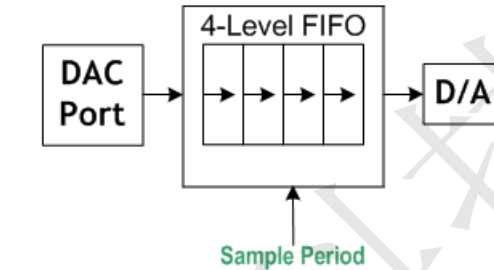


Figure 7.7 Auto FIFO Structure

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 or 128K 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.



CST1602B 16-bit Speech Processor

8. CST1602B Flash Control

8.1 Flash Structure

CST1602B is built in 128Kx16 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

CST1602B 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.



CST1602B 16-bit Speech Processor

8.4 Flash Word Read

CST1602B 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
```


CST1602B 16-bit Speech Processor

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.

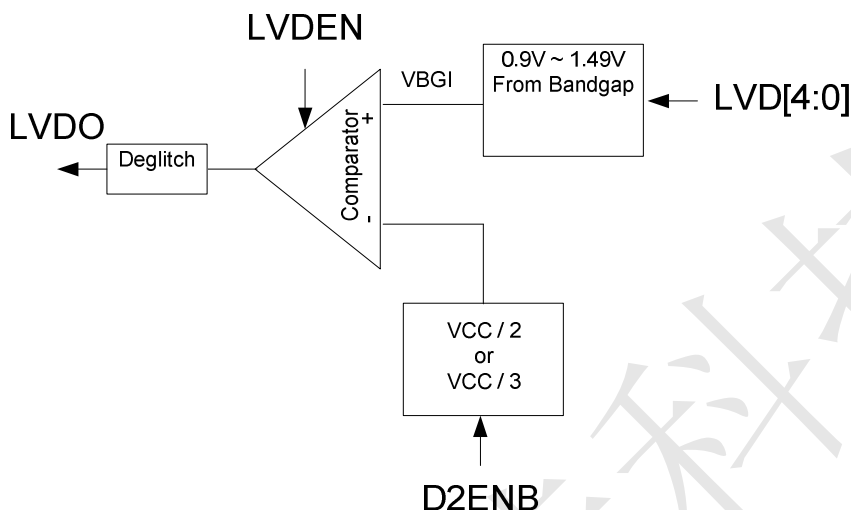


Fig 9.2-1 Function diagram of LVD

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

LVD [4:0]	D2 NB=0		D2 NB=1		LVD [4:0]	D2 NB=0		D2 NB=1	
	Ideal	Real	Ideal	Real		Ideal	Real	Ideal	Real
0x00	1.80	1.79	2.70	2.68	0x10	2.44	2.40	3.66	3.59
0x01	1.84	1.83	2.76	2.74	0x11	2.48	2.44	3.72	3.65
0x02	1.88	1.87	2.82	2.79	0x12	2.52	2.48	3.78	3.70
0x03	1.92	1.91	2.88	2.85	0x13	2.56	2.52	3.84	3.76
0x04	1.96	1.95	2.94	2.91	0x14	2.60	2.56	3.90	3.82
0x05	2.00	1.98	3.00	2.96	0x15	2.64	2.59	3.96	3.87
0x06	2.04	2.02	3.06	3.02	0x16	2.68	2.63	4.02	3.93
0x07	2.08	2.06	3.12	3.08	0x17	2.72	2.67	4.08	3.99
0x08	2.12	2.10	3.18	3.13	0x18	2.76	2.71	4.14	4.05
0x09	2.16	2.14	3.24	3.19	0x19	2.80	2.75	4.20	4.10
0x0A	2.20	2.18	3.30	3.25	0x1A	2.84	2.79	4.26	4.16
0x0B	2.24	2.21	3.36	3.30	0x1B	2.88	2.82	4.32	4.22
0x0C	2.28	2.25	3.42	3.36	0x1C	2.92	2.86	4.38	4.27
0x0D	2.32	2.29	3.48	3.42	0x1D	2.96	2.90	4.44	4.33
0x0E	2.36	2.33	3.54	3.48	0x1E	3.00	2.94	4.50	4.39
0x0F	2.40	2.37	3.60	3.53	0x1F	3.04	2.98	4.56	4.44

It's a COB test result for reference only.



CST1602B 16-bit Speech Processor

10. CST1602B System Control

10.1 Halt Mode & Wake up

CST1602B 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~PA3, PB0~PB5 and PC0~PC7 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.

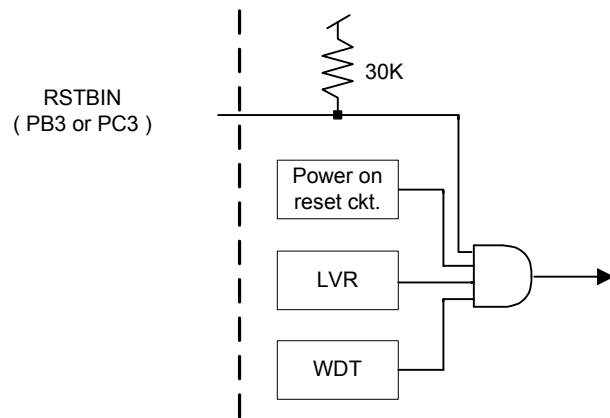
The reset watch dog timer code syntax is strongly recommended as: **"CLRWDT = AR"**.

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

CST1602B 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.





CST1602B 16-bit Speech Processor

10.6 Ultra Deep Sleep

- Ultra deep sleep mode only support PortA0~ PortA3 rising edge wake up
- Ultra deep sleep mode only supports execution in bank0
- All power off except necessary in Ultra deep sleep mode
- Typical standby current below 2uA in Ultra deep sleep mode
- Keep all I/O status in Ultra deep sleep mode

10.6.1 Control Registers

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
DEEP_SLEEP2.L	1FH	04	R/W	-	-	-	-	-	VDD1EN	CLK	DSENIN	Deep Sleep2 Status Register[7:0]
DEEP_SLEEP2.H	1FH	00	R/W	DSENO	-	-	-	WAKE[3:0]				Deep Sleep2 Status Register[15:8]

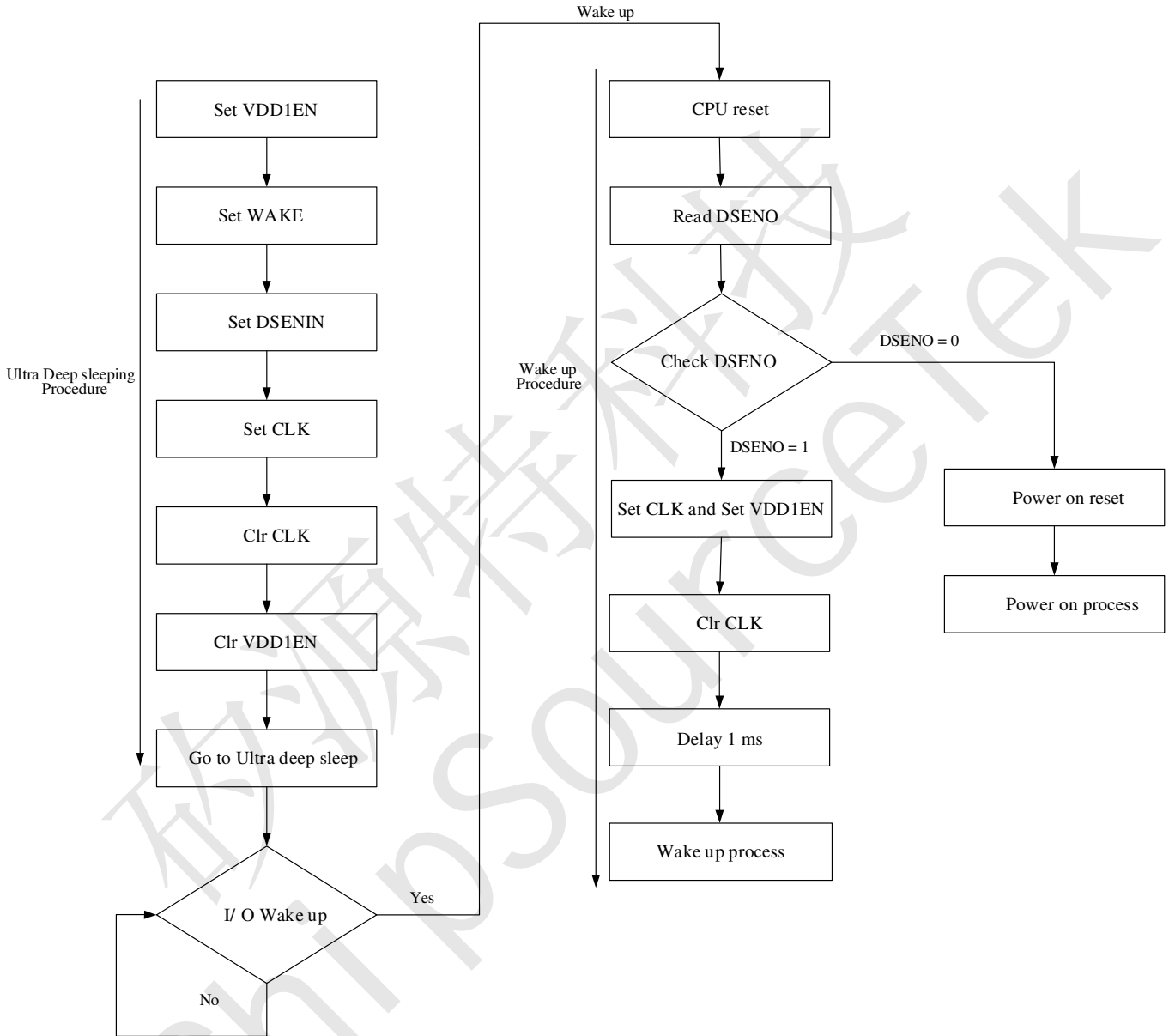
DSENIN	Ultra deep sleep input enable
CLK	Clock pulse generator
VDD1EN	1: Internal power on 0: Internal power off
WAKE	Default:0, Wake Up Enable Bit0: PortA0 wake up enable bit Bit1: PortA1 wake up enable bit Bit2: PortA2 wake up enable bit Bit3: PortA3 wake up enable bit Note: At least one wake up enable bit must be enabled.
DSENO	Ultra deep sleep wake up flag, 1: wake up from Ultra deep sleep mode occurred

Notice: Ultra deep sleep mode is invalid in ICE debug mode.



CST1602B 16-bit Speech Processor

10.6.2 Flow Chart





CST1602B 16-bit Speech Processor

10.6.3 Example Code

Ultra_deep_sleep_procedure:

```
ax = 0 // initial value

ar = IOP_DEEP_SLEEP2
IO[IOP_IX] = ar
set ax.b2 // Internal power on
ar = ax
IO[IOP_DAT] = ar

ar = IOP_DEEP_SLEEP2
IO[IOP_IX] = ar
set ax.b8 // PortA0 wake up enable
ar = ax
IO[IOP_DAT] = ar

ar = IOP_DEEP_SLEEP2
IO[IOP_IX] = ar
set ax.b0 // Ultra deep sleep input enable
ar = ax
IO[IOP_DAT] = ar

ar = IOP_DEEP_SLEEP2
IO[IOP_IX] = ar
set ax.b1 // Generate clock pulse high
ar = ax
IO[IOP_DAT] = ar

ar = IOP_DEEP_SLEEP2
IO[IOP_IX] = ar
clr ax.b1 // Generate clock pulse low
```




CST1602B 16-bit Speech Processor

ar = ax

IO[IOP_DAT] = ar

ar = IOP_DEEP_SLEEP2

IO[IOP_IX] = ar

clr ax.b2 // Internal power off

ar = ax

IO[IOP_DAT] = ar

ChipSourceTek



CST1602B 16-bit Speech Processor

Wake_up_procedure:

```
ar      = IOP_DEEP_SLEEP2
IO[IOP_IX]  = ar

ar = IO[IOP_DAT]
test ar.b15 // Read Ultra deep sleep wake up flag DSENO
if nz jmp wake_up_from_ultra_deep_sleep
jmp      power_on_reset // Run power on process
```

wake_up_from_ultra_deep_sleep:

```
ax = 0 // initial value

ar      = IOP_DEEP_SLEEP2
IO[IOP_IX]  = ar
set ax.b1 // Generate clock pulse high
set ax.b2 // Internal power on
ar= ax
IO[IOP_DAT] = ar

ar      = IOP_DEEP_SLEEP2
IO[IOP_IX]  = ar
clr ax.b1 // Generate clock pulse low
ar =ax
IO[IOP_DAT] = ar
```

Call delay_1ms

Call wake_up_process



CST1602B 16-bit Speech Processor

11. CST1602B 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	6.553MHz ±3%		13.107MHz ±3%	
RC oscillator frequency	Frc1		65.536MHz ±3%		
Low power RC oscillator frequency	Frc2	32768Hz -30%		32768Hz +30%	
Normal Sleep Halt Current	Ihalt1		5uA	9uA	All function off
Ultra Deep Sleep Halt Current	Ihalt2		2uA	3uA	VDD power 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

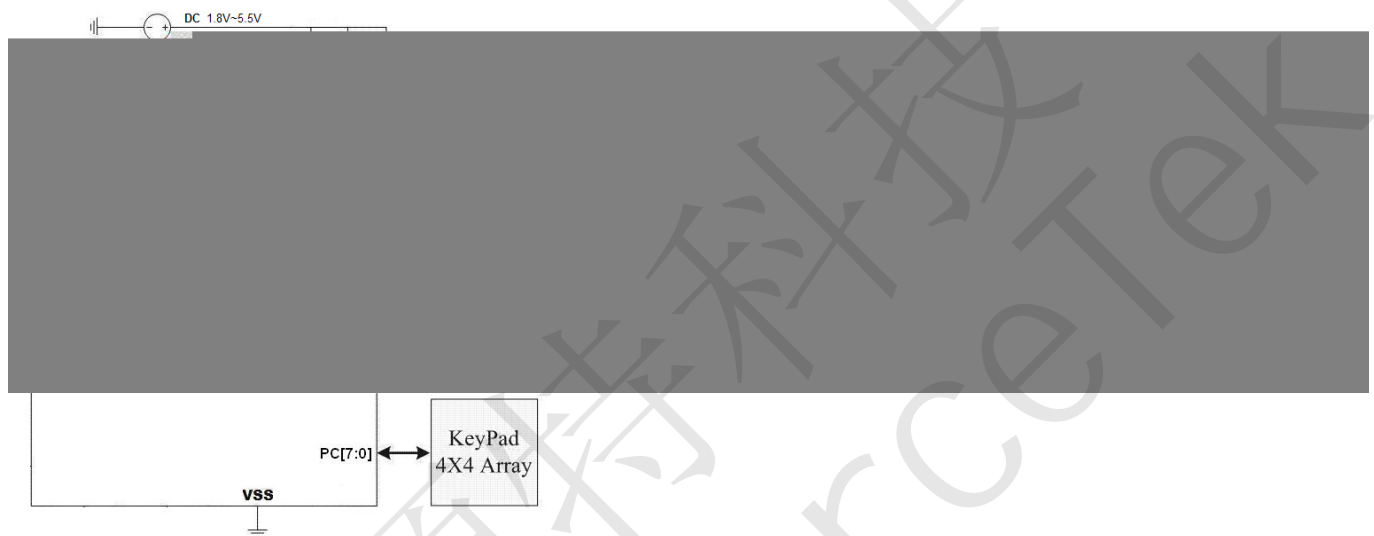
CST1602B 16-bit Speech Processor

12. CST1602B Application Circuit and Chip Power 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.





CST1602B 16-bit Speech Processor

13. CST1602B Appendix:

Appendix 1: PORT A PIN MAP RELATE TO FUNCTION

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

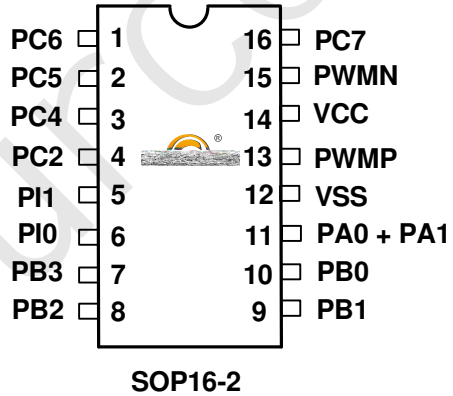
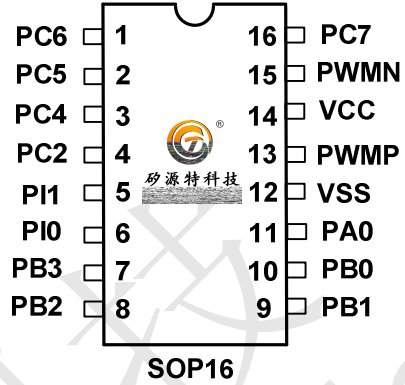
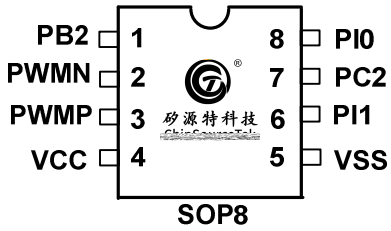
Appendix 3: PORT C PIN MAP RELATE TO FUNCTION

	PORT C[7:0]							
	7	6	5	4	3	2	1	0
SPI (Master)								
ICE								
IR 38K								
EXT-RESETB					EXT-RSTB(1)			
EXT-INT								



CST1602B 16-bit Speech Processor

14. CST1602B Package: SOP8 / SOP16 / SSOP24 / SOP16-2

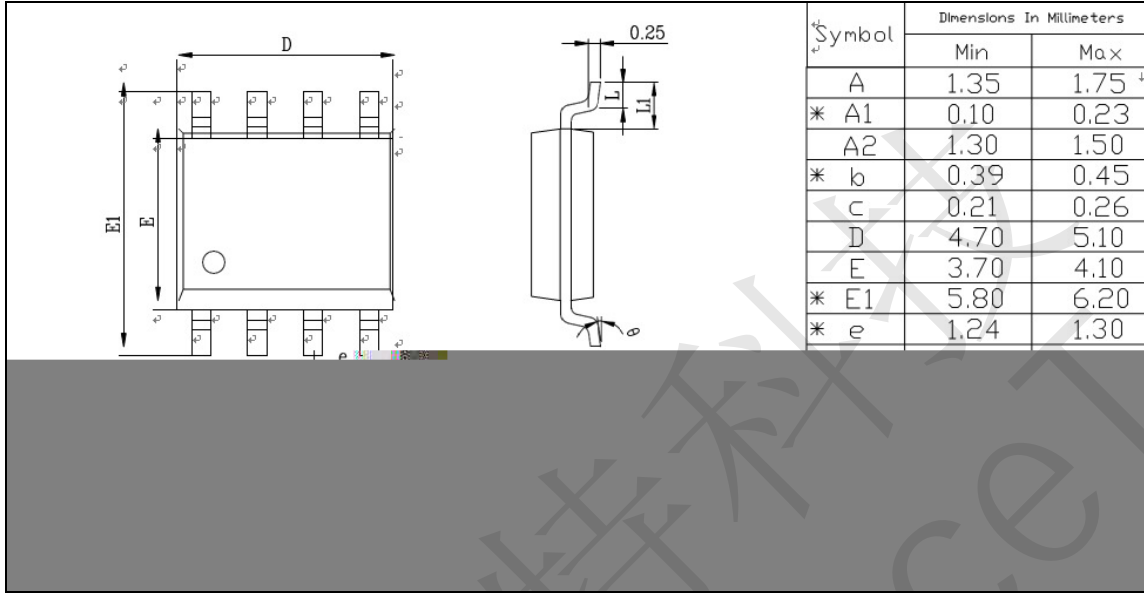




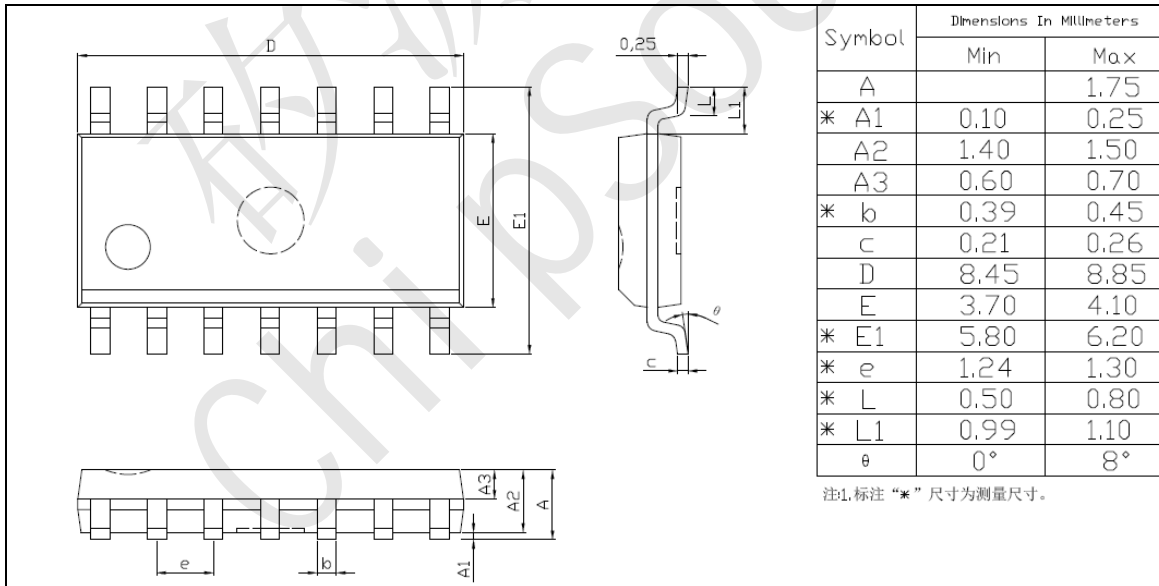
CST1602B 16-bit Speech Processor

PACKAGE INFORMATION

SOP08



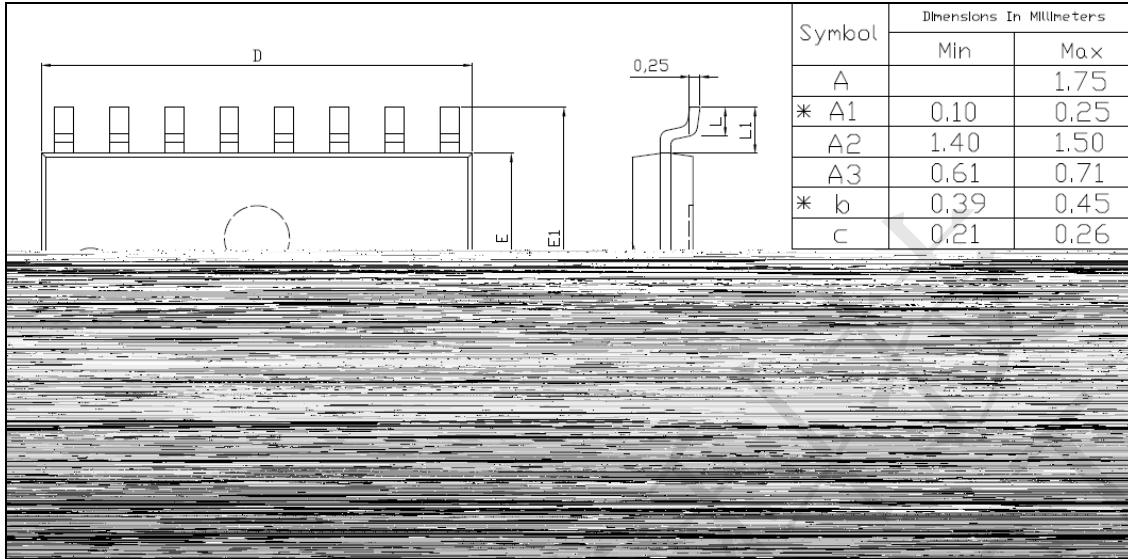
SOP14



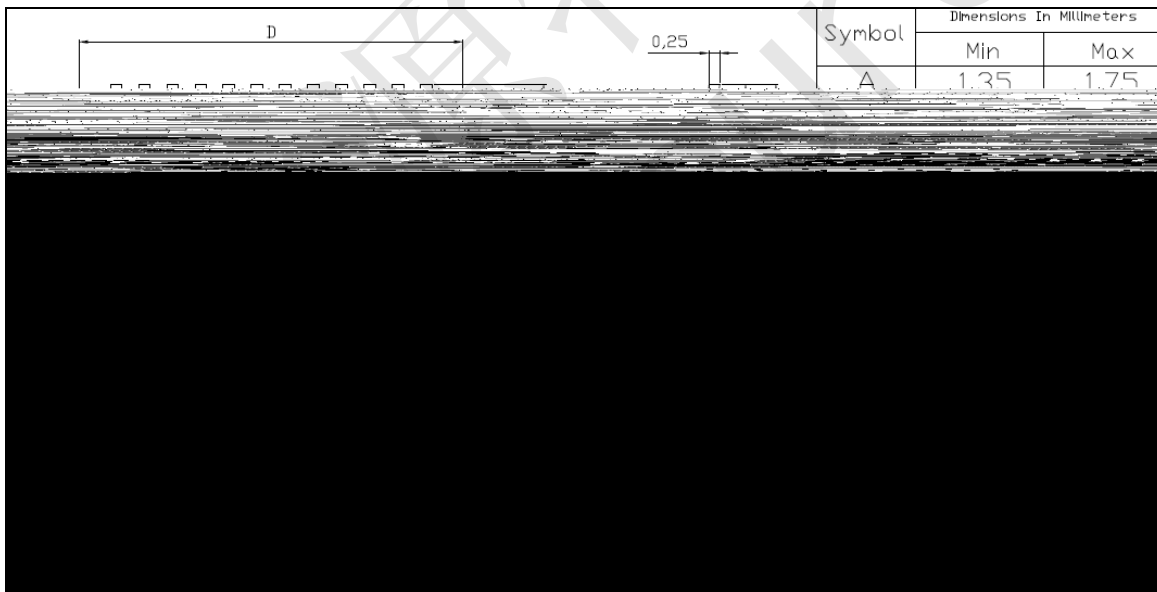


CST1602B 16-bit Speech Processor

SOP16

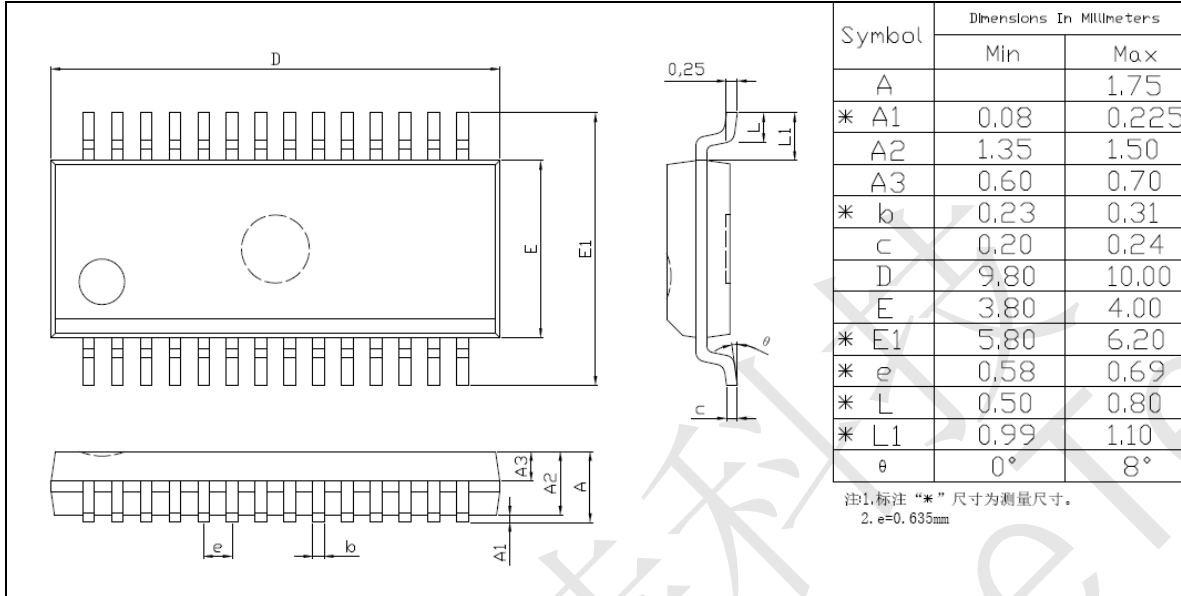


SSOP24



CST1602B 16-bit Speech Processor

SSOP28





CST1602B 16-bit Speech Processor

15. CST1602B 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
```



CST1602B 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
```



CST1602B 16-bit Speech Processor

16. CST1602B Revision history

REVISION	DESCRIPTION	PAGE	DATE
V1.0	New establish		2022/3/30

芯源特科技
ChipSourceTek