

JA23000

Product Guide

JAZTEK Technology, Inc.

Features

- Built-in an 8-bit CPU core
- Built-in 128~208 bytes of data RAM
- Built-in 0~1024KB user data ROM
- Built-in 16K~60KB program ROM
- On-chip RC oscillator or crystal oscillation clock input
- 32 KHz internal RC oscillator
- Illegal address reset
- Maximum of 24 programmable I/O pins
- Internal RC oscillator for Timer or WDT
- Watch-dog timer (WDT)
- Power saving STOP & HALT modes
- Maximum of 2*16-bit (or 8-bit) & 2*8-bit timers
- External interrupt input
- I/O State Change wake-up option for all of I/O port
- Operating voltage 2.4V – 5.4V
- Programmable R-option for PA4~7 & PB0~7
- Support 32768Hz crystal oscillator share with Port A
- 1ms interrupt (@ Fsys=2MHz)

General Description

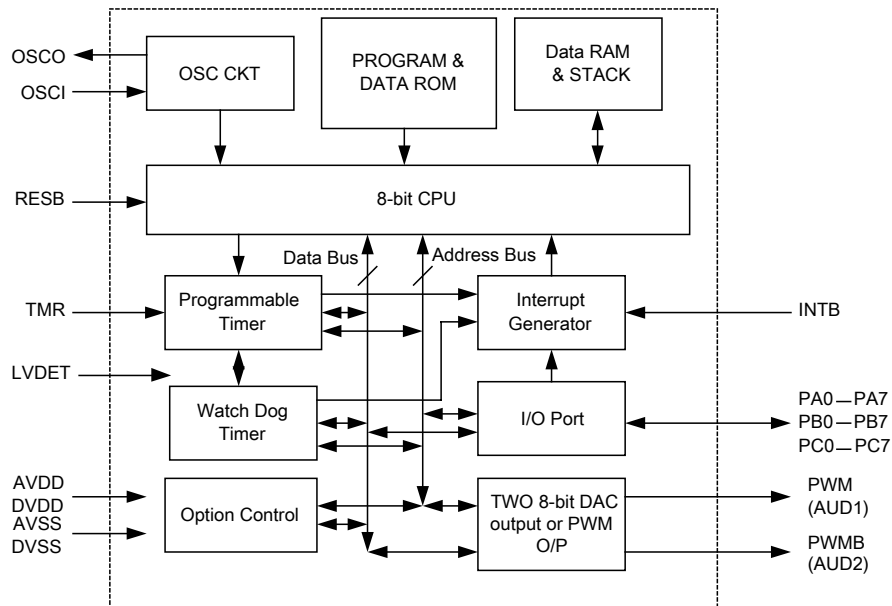
JA23000 is a series of 14 to 340 seconds (@S.R.=6kHz, 4-bit ADPCM) single chip voice synthesizer IC with a PWM Direct Drive circuit or AUD output for transistor application, JA23000 series contains an 8-bit Micro-Controller Units (MCU), 12~24 programmable general I/Os, 16KB~60KB Program ROM, Special register, 0~1024KB Data ROM, 4 CHs of current type DAC or 2 CHs PWM output for voice output, 128~208

bytes of user data RAM and 2*16-bit (or 8-bit) & 2*8-bit Timers. The advanced sub-micron CMOS process technology ensures JA23000 series high performance, high reliability and sophisticated functionalities. In addition, this chip also provides high sink current port pins, multi external interrupt pins, Low Voltage Detector (LVD) function, and multi oscillator options. JA23000 series offers the best cost/performance ratios, as a controller, for the industry applications.

Selection Table

Part No.	JA23048	JA23060	JA23384	JA23512	JA23768	JA231K0
I/O	12	12	24	24	24	24
PROM	48KB	60KB	16KB	16KB	16KB	16KB
DROM	0	0	384KB	512KB	768KB	1024KB
RAM	128B	128B	208B	208B	208B	208B
Timer (Bits * No.)	8b*2	8b*2	16*2 & 8*2	16*2 & 8*2	16*2 & 8*2	16*2 & 8*2
PWM	V	V	V	V	V	V
DAC		V	V	V	V	V

Block Diagram



Pin Assignment

Pin Name	I/O	Internal	Description
TMR	I/O	—	Timer counter input or timer clock output by pin option.
RESB	I	Pull Hi	System input, active low.
DVSS, AVSS	—	—	Negative power supply.
PA0 — PA7 PB0 — PB7 PC0 — PC7	I/O	Pull Hi	Port I/O pins can be configured as input, output, or interrupt input pin. Some pins can configure for special function. Please refer to the I/O configuration description.
DVDD, AVDD	—	—	Positive power supply.
OSCO	O	—	In XTAL mode, it connects to an external XTAL between OSCI and OSCO. In RC mode, its output is the $F_{sys}/4$.
OSCI	I	—	In XTAL mode, it connects to an external XTAL between OSCI and OSCO. In RC mode, it connects to an external oscillator resistor between OSCI and VDD. Meanwhile, the pin can be used as a external clock input.
INTB	I	Pull Hi	INT input pin by user program option.
PWM(AUD1), PWMB(AUD2)	O	PMOS (AUD1,2)	Current type DAC output for driving external transistors or driving speaker directly by mask option.
LVDET	I	—	Low voltage detection pin. Connect a resistor between LVDET and VSS to adjust the low battery detection.

Note :

1. The pull high with or without is optional by user program.
2. The AUD1 and AUD2 are shared with PWM and PWMB

Memory

- Memory Mapping

Address	Definition
00h	POWERC (W)
01h	INTC (R/W)
02h	INTF (R/W)
03h	WDTCLR (W)
04h	WDTC (R/W)
05h	TMR0H (R/W)
06h	TMR0L (R/W)
07h	TMR0C (R/W)
08h	TMR1H (R/W)
09h	TMR1L (R/W)
0Ah	TMR1C (R/W)
0Bh	PA (R/W)
0Ch	PAC (R/W)
0Dh	PAR (R/W)
0Eh	PB (R/W)
0Fh	PBC (R/W)
10h	PBR (R/W)
11h	PC(R/W)
12h	PCC(R/W)
13h	Not used
14h	Not used
15h	Not used
16h	Not used
17h	Not used
18h	Not used
19h	VOLC
1Ah	DA1 (WR)
1Bh	DA2 (WR)
1Ch	DA3 (WR)
1Dh	DA4 (WR)
1Eh	CROMCONT (W)
1Fh	CROMADD (R/W)
20h ~ 2Ah	Special RAM reserved for further expanding for setting internal as internal register
2Bh	TIMER2L (R/W)
2Ch	TIMER2C (R/W)
2Dh	TIMER3L (R/W)
2Eh	TIMER3C (R/W)
30h ~ FFh	General purpose Data Memory & Stack
100h ~ 0FFFh	Reserved used
1000h ~ FFFFh	User Program (12K~60KB) Note: Max 0.5KB for testing program

- The low nibble of VOLC controls the AUD1 & the high nibble of VOLC controls the AUD2 (DAC output only.)
- Data RAM
 - Total of 256 bytes of RAM (including the stack, and special register) is available from \$00h to \$FFh. The stack begins at the address \$FFh and proceeds down to \$00h.
 - For JA23048 & JA23060 : 128B (80h – FFh)
 - For JA23384 – JA231K0 : 208B (30h – FFh)
 - The address of 00h – 2Fh are designed for special registers.
- Program ROM
 - The JA23000 series includes the Max. of 60K bytes of user ROM which located from \$1000h to \$FFFFh. The 1K bytes of internal test ROM is for testing program. The PROM location of various bodies is presented as follows:
 - For JA23048 (48KB) : 4400h – FFFFh (4000h – 43FFh for testing program)
 - For JA23060 (60KB) : 1400h – FFFFh (1000h – 13FFh for testing program)
- For JA23384 – JA231K0 (16KB) : C400h – FFFFh (C000h – C3FFh for testing program) Data ROM
 - To write the address expander control register of CROMCONT, CROMADD can access max. of 1024KB data. The address from \$00000 to \$FFFFFF. The DROM location of various bodies is illustrated as follows :

Body	JA23384	JA23512	JA23768	JA231K0
DROM size	384KB	512KB	768KB	1024KB
DROM Address	00000h 5FFFFh	00000h 7FFFFh	00000h BFFFFh	00000h FFFFFh

- Reset, IRQ vectors
 - The address of RESET and IRQ are located from \$FFFC to \$FFFFh. The interrupt vectors should be specified in the program as follows :

```

ORG $FFFC
JMP RESET_VECTOR      ;($FFFC, $FFFD)
JMP INT_VECTOR       ;($FFFE, $FFFF)
```

Note : NMI (Non-Maskable Interrupt) is invalid in JA23000 Body.

- Stack Pointer
 - The stack pointer is set from \$FFh after power on.

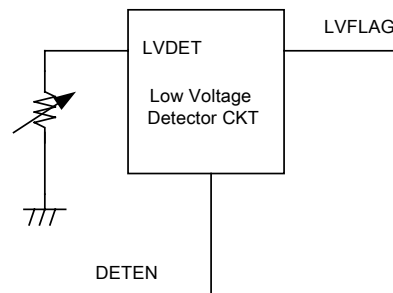
Power configuration

The JA23000 series provide two power saving mode, one is the HALT and the other is the STOP mode. When writing “1” to the HALT bit, system will enter HALT mode and system clock will stop whereas the internal RC will free run continuously. The timer overflow, WDT overflow, and external interrupt will wakeup the system. When writing “ 1” to the STOP bit, system will enter STOP mode and system clock, internal RC oscillator will be stop. Only external interrupt (INT or PX change state) can wakeup the system. When system is overflow, the HALT and STOP bit will be cleared to “0” automatically.

When writing “1” to the DADIS bit, the DAC output will be disabled.

When the DETEN bit is written as “1”, then the low battery detector circuit will be enabled, and if low battery is detected (set by external R), LVFLAG will be set to “1”. Otherwise, this bit is set to “0”. After writing the DETEN bit, user must insert 2 NOP instructions, and then read the stable LVFLAG data.

Note: DADIS function is invalid in JA23384.JA23512.JA23768 and JA231k0.



POWERC

Register	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POWERC	00h	LVFLAG	DETEN	—	STOP	—	—	DADIS	HALT

In HALT mode, the 32K(PA or WDT) is On & Fosc is OFF. In STOP mode, the 32K(PA or WDT) & Fosc are all OFF.

I/O configuration

Maximum of 24 I/Os (grouped into 3 I/O ports, PA, PB, and PC) are provided. All of these I/O ports can be used as input and output operations. For input operation, these ports are non-latched and for output operation all the data are latched and remain unchanged till the output latch is re-written.

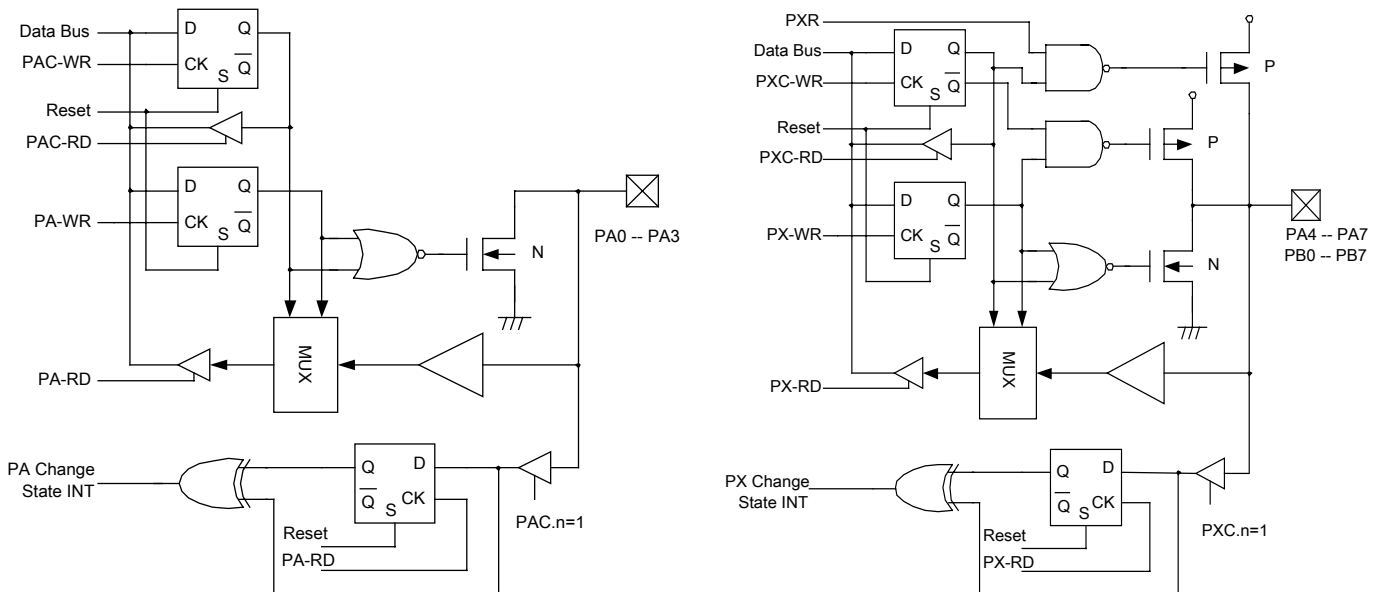
Each I/O line has its own control register (PAC, PBC, and PCC) to control the input/output configuration. If the global interrupt (INTC.0) is disabled and the corresponding I/O interrupt (INTC.4; INTC.5 & INTC.6) is enabled, it will activate the system and its corresponding flag will be set when the corresponding I/O state is changed.

Part No.	I/O	Port A	Port B	Port C
JA231K0	24	8	8	8
JA23768	24	8	8	8
JA23512	24	8	8	8
JA23384	24	8	8	8
JA23064	12	8	4	
JA23048	12	8	4	

PA Configuration

Label	Address	Function	R/W	Default
PA	0Bh	PA data input/output	R/W	FF
PAC	0Ch	PA direction control, 1=input 0=output	R/W	FF
PAR	0Dh	PA pull-high resistor option, 1=With, 0=Without	R/W	FX

1. The PA0 – PA3 they are defined as NMOS open drain output only when PAC is set as output mode.
2. The PA0 – PA3 without pull-high resistor in input mode.



PB Configuration

Label	Address	Function	R/W	Default
PB	0Eh	PB data input/output	R/W	FF
PBC	0Fh	PB direction control, 1=input 0=output	R/W	FF
PBR	10h	PB pull-high resistor option, 1=With, 0=Without	R/W	FF

1. Its diagram is shown above. The output mode (PB0 — PB7) is the same as PA4 — PA7.
2. The pull-high resistor will be disabled automatically when port is programmed as output mode.

PC Configuration

Label	Address	Function	R/W	Default
PC	11h	PC data input/output	R/W	FF
PCC	12h	PCC direction control, 1=input 0=output	R/W	FF

1. The internal pull high resistor of PC can be enabled or disabled by mask option.
2. The pull-high resistor will be disabled automatically when port is programmed as output mode.

Timer

There are 4 sets of programmable timer in JA23000 series, they are TMR0 (05h, 06h & 07h), TMR1 (08h, 09h & 0Ah), TMR2 (2Bh & 2Ch) & TMR3 (2Dh & 2Eh).

Part No.	TMR0	TMR1	TMR2	TMR3
JA231K0	8-bit/16-bit	8-bit/16-bit	8-bit	8-bit
JA23768	8-bit/16-bit	8-bit/16-bit	8-bit	8-bit
JA23512	8-bit/16-bit	8-bit/16-bit	8-bit	8-bit
JA23384	8-bit/16-bit	8-bit/16-bit	8-bit	8-bit
JA23064	8-bit	8-bit		
JA23048	8-bit	8-bit		

Note : The TMR0 and TMR1 can be selected as 16-bit or 8-bit mode by mask option.

TMR0H, TMR0L (05h, 06h) & TMR0C (07h)

The TMR0 is a programmable 16-bit/8-bit count-up counter. The counter registers are named as TMR0L (06h) & TMR0H (05h). The clock source may come from Fosc/4, internal RC clock or external. The TMR0C is its control register and the default value is 00. The definition of TMR0C is listed below.

Labels	Bits	Function
TON0 (TMR0)	0	Timer0 enable/disable definition bit 0 = Disable; 1 = Enable
TS0, TS1, TS2	1 - 3	Timer clock source selection bits
—	4	Reserved, not used
TMR/WDT	5	To define the clock source is with pre-scale counter 0: Timer 1: WDT
TM0 TM1	6, 7	To define the operation mode 00 = Timer mode (internal system clock; Fosc/4) 01 = Timer mode (internal RC clock) 10 = Event count mode (external clock) 11 = Timer mode (internal system clock; Fosc or 32768 Hz option by X12PA)

- Notes :
- The event counter behaves from low to high transition.
 - When the X12PA=0 and (TM0, TM1)=(1, 1), then the clock source is 32KHz.
 - When the X12PA=1 and (TM0, TM1)=(1, 1), then the clock source is the Fosc(4MHz).
 - X12PA is a configuration option by user (code layer option).

TS2	TS1	TS0	TMR Rate	WDT Rate
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

TMR1H, TMR1L (08h, 09h) & TMR1C (0Ah)

The TMR1 is a programmable 16-bit/8-bit count-up counter. The counter registers are named TMR1L (09h) and TMR1H (08h) respectively. The clock source is defined by the register of TMR1C. The TMR1C is its control register and the default value is 00. The definition of TMR1C is listed below.

Labels	Bits	Function
TON1	0	Timer1 enable/disable definition bit 0 = Disable; 1 = Enable
TS0, TS1, TS2	1 - 3	Timer clock source selection bits
—	4	Reserved, not used
TMR	5	The set the TMR as input or output pin. 0=input 1=output, it will generate a clock output from TMR pin.
TM0 TM1	6, 7	To define the operation mode 00= Timer mode (internal system clock; Fosc/4) 01= Timer mode (internal RC clock) 10= Event count mode (external clock) 11= Timer mode (internal system clock; Fosc or 32768 Hz option by X12PA)

- Notes :
1. The event counter behaves from low to high transition.
 2. When the X12PA=0 and (TM0, TM1)=(1, 1), the clock source is 32KHz.
 3. When the X12PA=1 and (TM0, TM1)=(1, 1), the clock source is the Fosc(4MHz).
 4. X12PA is a configuration option by user (code layer option).
 5. When the TMR pin is set as output, the initial state is low.
 6. The TMR as output function is invalid in JA23384.JA23512.JA23768 and JA231k0.

TMR2L (2bh) & TMR2C (2ch)

The TMR2 is a programmable 8-bit count-up counter. The clock source comes from Fosc/4, internal RC clock or external. The TMR2C is its control register and the default value is 00. The definition of TMR2C is listed below.

Labels	Bits	Function
TON2	0	Timer2 enable/disable definition bit 0 = Disable; 1 = Enable
TS0, TS1, TS2	1 - 3	Timer clock source selection bits
—	4	Reserved, not used
—	5	Reserved, not used
TM0 TM1	6, 7	To define the operation mode 00= Timer mode (internal system clock; Fosc/4) 01= Timer mode (internal RC clock) 10= Event count mode (external clock) 11= Timer mode (internal system clock; Fosc or 32768 Hz option by X12PA)

- Notes :
1. The event counter behaves from low to high transition.
 2. When the X12PA=0 and (TM0, TM1)=(1, 1), the clock source is 32KHz.
 3. When the X12PA=1 and (TM0, TM1)=(1, 1), the clock source is the Fosc(4MHz).
 4. X12PA is a configuration option by user (code layer option).

TMR3L (2dh) & TMR3C (2eh)

The TMR3 is a programmable 8-bit count-up counter. The clock source comes from Fosc/4, internal RC clock or external. The TMR3C is its control register and the default value is 00. The definition of TMR3C is listed below.

Labels	Bits	Function
TON3	0	Timer3 enable/disable definition bit 0 = Disable; 1 = Enable
TS0, TS1, TS2	1 - 3	Timer clock source selection bits
—	4	Reserved, not used
—	5	Reserved, not used
TM0 TM1	6, 7	To define the operation mode 00= Timer mode (internal system clock; Fosc/4) 01= Timer mode (internal RC clock) 10= Event count mode (external clock) 11= Timer mode (internal system clock; Fosc or 32768 Hz option by X12PA)

- Notes :
1. The event counter behaves from low to high transition.
 2. When the X12PA=0 and (TM0, TM1)=(1, 1), the clock source is 32KHz.
 3. When the X12PA=1 and (TM0, TM1)=(1, 1), the clock source is the Fosc(4MHz).
 4. X12PA is a configuration option by user (code layer option).

TS2	TS1	TS0	TMR Rate
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

In timer mode, user should write an initial value to the counter registers (TMRnL, TMRnH) and then enable the TONn (TMRnC.0). The TMRn will count-up with the rate of TS0 – TS2 & TM0, TM1 setting. When timer is overflow (ffffh --> 0000h), system will generate an interrupt and the corresponding flag will be set to “1”, if both the global interrupt bit and the corresponding bit are enabled. After timer is overflow, the initial value will reload to counter register automatically unless the TMRn is disabled.

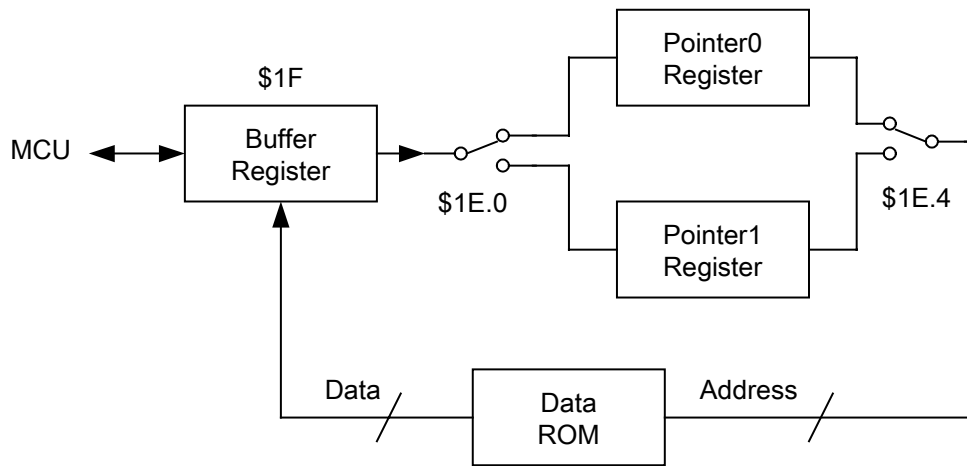
In event count mode, counter registers will count up a step when the TMR pin inputs a changing state from “L” to “H”. When the timer is overflow and the corresponding control bits are set, the system will also generate an interrupt.

Data Access (1Eh, 1Fh)

For JA23048 & JA23060, data is accessed by index addressing instruction. So the CROMRCONT (1Eh) and CROMADD (1Fh) registers are useless.

For JA23384~JA231K0 bodies, data is accessed by CROMCONT & CROMADD to addressing and read data.

ADDRESS	Name
1Eh	CROMCONT (W)
1Fh	CROMADD(R/W)



- Write Initial Address**

There are 2 addressing pointers for the data memory (Pointer0, Pointer1). And at the initial state, the host controller should select one of the pointers to write the start address, which is defined by \$1E.0. The start address must be 3 bytes (ADDR_L, ADDR_M, ADDR_H). If over 3 bytes of data written to the pointer, then the 1st position is written again. After 3 bytes of start address are written, then the host controller can read the data from \$1F register.

Code(\$1E.0)	Pointer	Default
0	0	V
1	1	

- Read Data**

After 3 bytes of initial address are written down, then the host controller can read data from \$1F, which is defined by the \$1E.4. And the read command will reset the address counter of corresponding pointer to ADDR_L. After read the \$1F one time then the corresponding pointer address will increment automatically.

Code(\$1E.4)	Pointer	Default
0	0	V
1	1	

- For JA23048 & JA23060 example**

The start address: \$3EC0

Program:

```
ADDRL: equ $1A
```

```
ADDRH: equ ADDR_L+1
```

; To declare the ADDR_L, ADDR_H location, and their location must be continuous.

```
...
lda #C0
sta ADDL
```

```

lda  #$3E
sta  ADDH
ReadLoop:
lda  #0          ; Clear the X register to zero
lda  (ADDRL,x)  ; Read the data from the start address
sta  BUFFER     ; Save the read data to Buffer register
inc  ADDRL
bne  +2         ; If ADDRL = 0, then escape the next instruction
inc  ADDRH     ; If ADDRL=0, then generate a carry to ADDRH
...           ; To process the read data
jmp  ReadLoop

```

- **For JA23384– JA231K0 bodies example**

- a. One pointer application

The start address: \$3EC0 and using pointer0

Program:

```

...
lda  #00
sta  $1E      ; Set the pointer0 to addressing & read
lda  $1F      ; Reset the address counter
lda  #$C0    ; ADD_L data
sta  $1F
lda  #$3E    ; ADD_M data
sta  $1F
lda  #$00    ; ADD_H data
sta  $1F
nop
nop          ; To delay 2 nop instruction
lda  $1F     ; Read the $3EC0 data, and the address pointer change to $3EC1
....       ; Data process
lda  $1F     ; Read $3EC1 data, , and the address pointer change to $3EC2
...

```

- b. Two pointers application

Section1 start address: \$003EC0 and using pointer0 to read

Section2 start address: \$01CF00 and using pointer1 to read

Program:

```

...
lda  #00
sta  $1E      ; Set the pointer0 to addressing & read
lda  $1F      ; Reset the address counter
lda  #$C0    ; ADD_L data of pointer0
sta  $1F
lda  #$3E    ; ADD_M data of pointer0
sta  $1F
lda  #$00    ; ADD_H data of pointer0
sta  $1F
,*****
lda  #01
sta  $1E      ; Set the pointer1 to addressing & pointer0 to read
lda  $1F      ; Reset the address counter
lda  #$00    ; ADD_L data of pointer1
sta  $1F
lda  #$CF    ; ADD_M data of pointer1
sta  $1F
lda  #$01    ; ADD_H data of pointer1

```

```

sta $1F
;*****
,
lda $1F ; Read the $3EC0 data, and the address pointer change to $3EC1
... ; Data process
lda $1F ; Read $3EC1 data, , and the address pointer change to $3EC2
...
;*****
,
lda #$10
sta $1E ; Set pointer1 to read & pointer0 to addressing
lda $1F ; Read the $01CF00 data, and the address pointer change to $01CF01
... ; Data process
lda $1F ; Read $01CF01 data, , and the address pointer change to $01CF02
...

```

Watchdog Timer

WDTCLR (W)

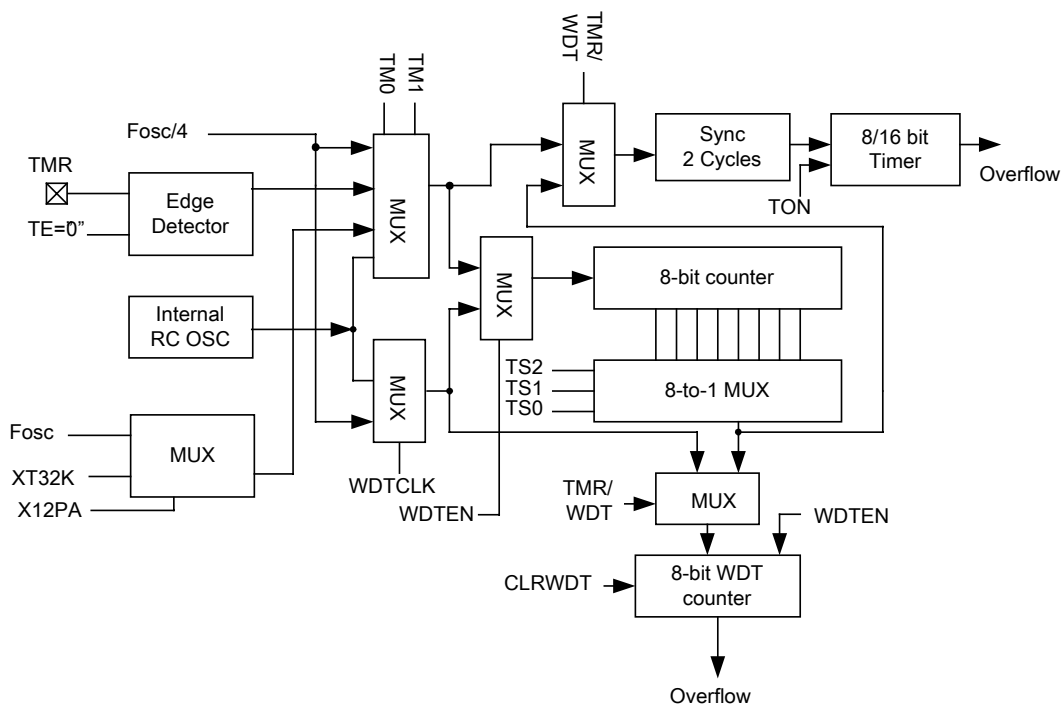
Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
03h		—	—	—	—	—	—	CLRWDT

Bit 0 (CLRWDT): To prevent a WDT time-out, reset is done by writing a “1” to this bit within a specific time WDTC (R/W)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
04h	—	—	—	WDTCLK	—	—	—	WDTEN

Bit 4 (WDTCLK): To select the WDT clock source
 0=Fosc/4 (Default), 1= Internal RC clock

Bit 0 (WDTEN): To enable/disable the WDT
 0= Disable (Default), 1= Enable



Interrupt

When an interrupt is generated, the interrupt vector (\$FFFE) will be loaded to PCL and PCH (PCL, PCH: Program Counter) and the original data in PCL, PCH & Status register content will be saved in stack. The corresponding interrupt bit (INTF.X) will be set (Write in "1") automatically. After the instruction "RTI" is executed, the original data will be pulled out from stack and saved to original registers. When the interrupt program is executed, the corresponding bit of interrupt flag (INTF.X) should be cleared (Write in "0") by user program.

The interrupt source include one of the following conditions:

- Timer overflow
- PA, PB, PC change state input
- External interrupt input

INTC (R/W)

Register	Bit No.	Label	Function
INTC	0	INTE	Global interrupt enable bit (1= Enabled; 0 = Disabled)
	1	INT	External INT pin interrupt Enable bit (1= Enabled; 0 = Disabled)
	2	TMR0	TMR0 interrupt Enable bit (1= Enabled; 0 = Disabled)
	3	TMR1	TMR1 interrupt Enable bit (1= Enabled; 0 = Disabled)
	4	PAI	Port A change state interrupt Enable bit (1= Enabled; 0 = Disabled)
	5	PBI	Port B change state interrupt Enable bit (1= Enabled; 0 = Disabled)
	6	PCI/1ms	Port C or 1ms change state interrupt Enable bit (1= Enabled; 0 = Disabled)
	7	TMR2/TMR3	TMR2 or TMR3 interrupt Enable bit (1= Enabled; 0 = Disabled)

INTF (R/W)

Register	Bit No.	Label	Function
INTF	0	INTF	External INT interrupt flag bit (1= Active; 0 = Inactive)
	1	TMR0F	TMR0 timer interrupt flag bit (1= Active; 0 = Inactive)
	2	TMR1F	TMR1 timer interrupt flag bit (1= Active; 0 = Inactive)
	3	TMR2F	TMR2 timer interrupt flag bit (1= Active; 0 = Inactive)
	4	PAF	Port A change state interrupt flag bit (1= Active; 0 = Inactive)
	5	PBF	Port B change state interrupt flag bit (1= Active; 0 = Inactive)
	6	PCF/1ms	Port C or 1ms change state interrupt flag bit (1= Active; 0 = Inactive)
	7	TMR3F	TMR2 or TMR3 timer interrupt flag bit (1= Active; 0 = Inactive)

Note : 1. The time period of 1ms is generated when system frequency is 2MHz.

2. The INTF can write-in "0" only. The data of "1" is ineffective. If want to clear INTF, don't use the command "AND", must write-in "0" to INTF immediately, see the Example for INTF process.
3. After the interrupt program is executed, the corresponding flag should be cleared by software.
4. If using port change-state interrupt, be sure to read port state before returning from interrupt subroutine. Without reading port state again, system might be unstable.

Example for INTF process

INTREQUEST:

```
sei
pha
txa
pha
lda INTF ;check TMR0 interrupt
bit #$02
beq OtherINT
jsr TMR0_interrupt
```

OtherINT:

```
...
pla
tax
pla
cli
rti
```

TMR0_interrupt:

```
lda #$FD ;clear TMR0 interrupt flag;INTF only write "0"
sta INTF
...
rts
```

Example for PB interrupt process

Program:

```
...
lda PB ;before set PB interrupt flag must read PB
lda #$21 ;set PB interrupt flag
sta INTC
...
```

INTREQUEST:

```
pha
txa
pla
lda #$01
bit INTF
...
lda #$20
bit INTF
beq OtherINT
jsr PB_interrupt
```

OtherINT:

```
...
pla
```

```
tax
pla
cli
rti
```

PB_interrupt:

```
lda  #0DF          ;INTF only write 0, can't write 1
sta  INTF
lda  PB            ; Port interrupt must read again before leave
...
rts
```

Wake up

The wake-up source include one of the following condition:

- Timer overflow
- PA, PB, PC change state input, if use port to wake up, must read the port before power down.
- External interrupt input

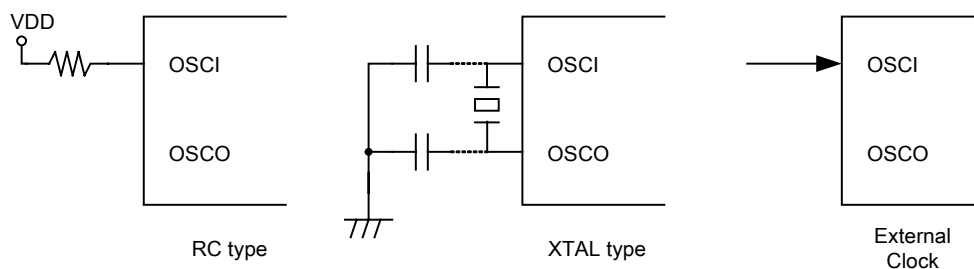
When the global interrupt bit (INTE) is disabled and the corresponding I/O port (PA, PB & PC) interrupt bit is enabled, the wakeup source will generate wakeup without interrupt.

After the interrupt is generated, the system will be activated from HALT mode or the STOP mode and then the program counter will increase and the next instruction will be executed continuously. All of the content of registers will be unchanged.

Oscillator

The JA23000 SERIES supports three types of oscillation circuit; they are RC, XTAL1 (High frequency) and XTAL2 (low frequency). For the RC type, an external resistor connection between OSC1 and VDD. For the crystal/resonator type, one of the crystal or resonator is connected to OSC1 and OSC0. For the crystal mode, it can be optioned as high frequency and low frequency operating mode.

For the XTAL mode, the system will delay 1024 clocks after power is turned on or the system is waken-up from the stop mode. And for the RC mode, the system will delay 64 clocks.



Reset

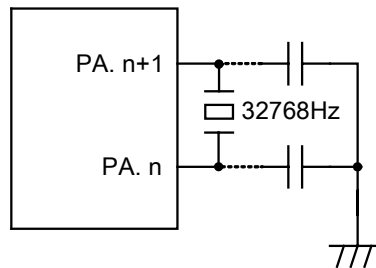
There are 5 conditions will reset the system

- Power on reset
- Reset pin active
- Illegal address generation
- WDT overflow
- VDD voltage lower than 1.8V

When system is reset, the reset vector (\$FFFC, \$FFFD) will load to PCL, PCH and instructions in the vector will be executed immediately. Only the power-on reset (cold reset) can initialize the internal register, the others reset (warm reset) can't change the content of all registers.

PA & 32768 generator

The PA port can share 2 pins to generate 32K frequency for internal timer by mask option.



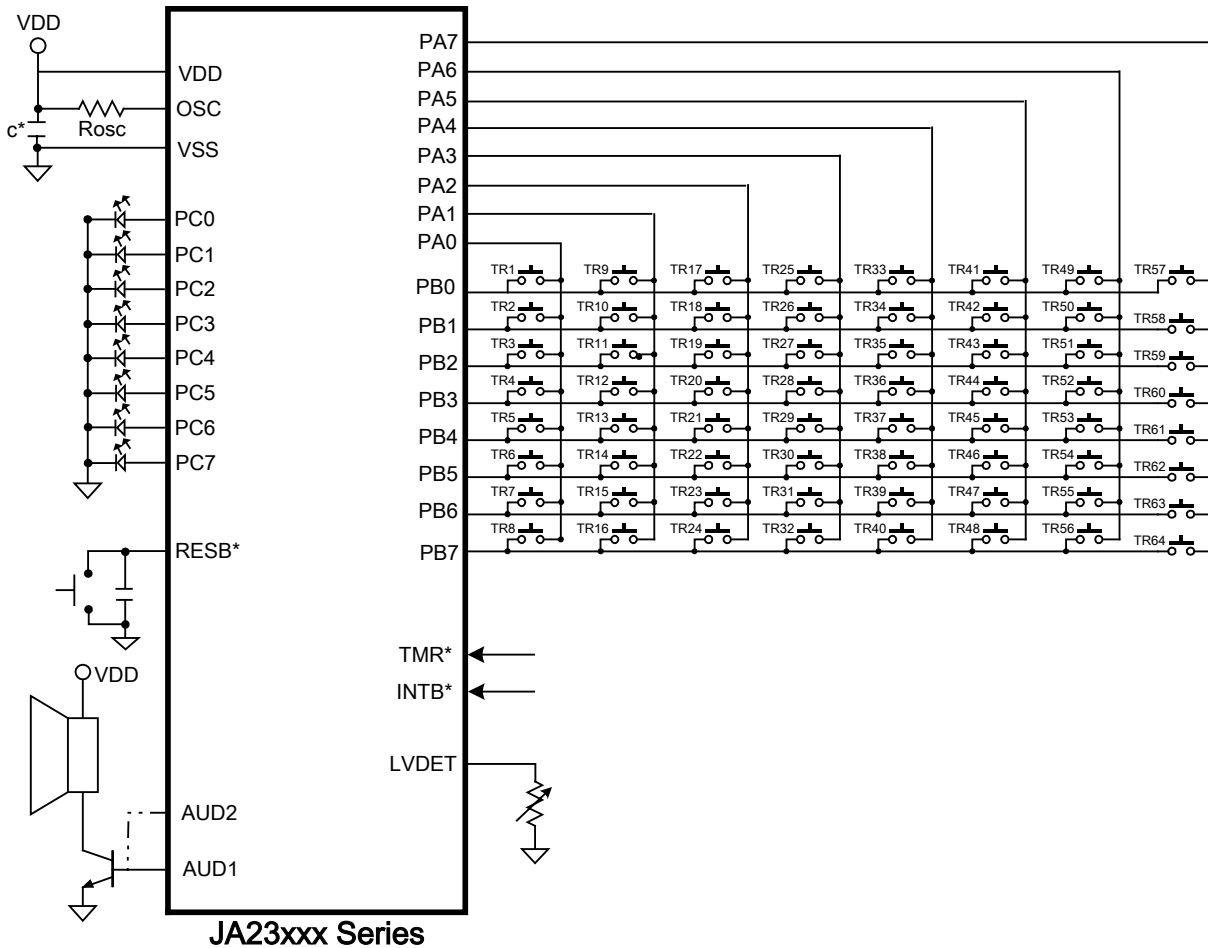
Mask Option

- a. TMR0 & TMR1 length: (16-bit or 8-bit)
- b. AUD1, AUD2 output: (DAC, PWM)
- c. PC pull high: (With, Without)
- d. PA pin option: (I/O, 32K oscillator)
- e. PC/1mS interrupt source : (PC port, 1mS)

Register Initial Summarized

Register	Address	Power on reset	Reset (WDT Overflow, RESB Active, Illegal address)	Reset (WDT overflow from Halt)
INTC	01h	0000 0000	0000 0000	uuuu uuuu
INTF	02h	0000 0000	0000 0000	uuuu uuuu
WDTCLR	03h	-----	-----	uuuu uuuu
WDTC	04h	--- 0 --- 0	--- 0 --- 0	uuuu uuuu
TMRH	05h/08h	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMRL	06h/09h	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMRC	07h/0Ah	0000 0000	0000 0000	uuuu uuuu
PA, PB, PC	0Bh/0Eh/11h	1111 1111	1111 1111	uuuu uuuu
PAC, PBC, PCC	0Ch/0Fh/1Dh	1111 1111	1111 1111	uuuu uuuu

Application Circuit



Note : 1.The IC substrate should be connect to VSS

2. The RESB, INTB, &TMR should be connected to VDD, if don't use.

3. The capacitor "c" is suggested as 0.1 μ F. (if with motor application, a 10-100 μ F capacitor should be added in addition)

Appendix

● Data Format

After JAICE compile the input data input and transfer to various format of data. The data format defines as follows :

a. The format of speech

1. 8-bit PCM data format

Addr0								Addr1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
System Clock				Type				Sample rate							
Addr2								Addr3							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Length (low byte)								Length (High byte)							
Addr4								Addr5							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Data0								Data1							

2. 4-bit ADPCM

Addr0								Addr1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
System Clock				Type				Sample rate							
Addr2								Addr3							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Length (low byte)								Length (High byte)							
Addr4								Addr5							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Data1				Data0				Data3				Data2			

3. 5-bit ADPCM

Addr0								Addr1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
System Clock				Type				Sample rate							
Addr2								Addr3							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Length (low byte)								Length (High byte)							
Addr4								Addr5							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Sign bit								Data1				Data0			

4. Silence data format

Addr0								Addr1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
System Clock				Type				Sample rate							
Addr2								Addr3							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Length (low byte)								Length (High byte)							

5. The format of raw data (binary code)

Addr0								Addr1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Data0								Data1							

6. End of section

Addr0							
b7	b6	b5	b4	b3	b2	b1	b0
\$F				\$F			

The System Clock code (b7 – b4)

System clock	1MHz	2MHz	4MHz	6MHz	8MHz
Code	0001	0010	0100	0110	1000

The type code (b3 – b0)

Type	Silence	4-bit (ADPCM)	8-bit (PCM)	5-bit (ADPCM)
Code	0000	0001	0010	1000

• Instruction Set
For JA23048 & JA23060

Instruction	Operation	Flag	Addressing Mode	Note
ADC	A+M+C A, C	N, Z, C, V	IMM, ZP	
AND	A M A	N, Z	IMM, ZP	
BCC	Jump if C=0	—	REL	
BCS	Jump if C=1	—	REL	
BEQ	Jump if Z=1	—	REL	
BIT	A M, M7 N, M6 V	N, Z, V	ZP, ABS	
BMI	Jump if N=1	—	REL	
BNE	Jump if Z=0	—	REL	
BPL	Jump if N=0	—	REL	
BVC	Jump if V=0	—	REL	
BVS	Jump if V=1	—	REL	
CLC	0 C	C=0	IMP	
CLI	0 I	I=0	IMP	
CLV	0 V	V=0	IMP	
CMP	A – M	N, Z, C	IMM, ZP, ZPX	
CPX	X – M	N, Z, C	IMM, ZP	
DEC	M–1 M	N, Z	ZP, ZPX	
DEX	X–1 X	N, Z	IMP	
EOR	A.XOR. M A	N, Z	IMM, ZP	
INC	M+1 M	N, Z	ZP	
INX	X+1 X	N, Z	IMP	
JMP	(PC+1) PCL, (PC+2) PCH	—	ABS, IND	
JSR	(PC+1) PCL, (PC+2) PCH PC+2 STACK	—	ABS	
LDA	M A	N, Z	IMM, ZP, ZPX, ABS, ABSX, INDX	
LDX	M X	N, Z	IMM, ZP	
NOP	—	—	IMP	

ORA	A.OR. M A	N, Z	IMM, ZP	
PHA	A STACK	—	IMP	
PHP	P (status) STACK	—	IMP	
PLA	STACK A	N, Z	IMP	
PLP	STACK P (status)	ALL	IMP	
ROL	C ACC or M C	N, Z, C	ACC, ZP	
ROR	C ACC or M C	N, Z, C	ACC, ZP	
RTI	STACK P STACK-1 PCH STACK-2 PCL	ALL	IMP	
RTS	STACK PC PC+1 PC	—	IMP	
SBC	A-M- C A	N, Z, C, V	IMM, ZP	
SEC	1 C	C=1	IMP	
SEI	1 I	I=1	IMP	
STA	A M	—	ZP, ZPX, INDX	
STX	X M	—	ZP, ABS	
TAX	A X	N, Z	IMP	
TSX	S X	N, Z	IMP	
TXA	X A	N, Z	IMP	
TXS	X S	—	IMP	

Note : For the JA23048 & JA23064, instructions of Y index or address mode with Y index are useless.

For JA23384 – JA231K0 bodies

Instruction	Operation	Flag	Addressing Mode	Note
ADC	A+M+C A, C	N, Z, C, V	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
AND	A M A	N, Z	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
ASL	C b7---b0 0	N, Z, C	ACC, ZP, ZPX, ABS, ABSX	
BCC	Jump if C=0	—	REL	
BCS	Jump if C=1	—	REL	
BEQ	Jump if Z=1	—	REL	
BIT	A M, M7 N, M6 V	N, Z, V	IMM,ZP,ZPX, ABS,ABSX	
BMI	Jump if N=1	—	REL	
BNE	Jump if Z=0	—	REL	
BPL	Jump if N=0	—	REL	
BRK	Force interrupt	B, I	IMP	
BVC	Jump if V=0	—	REL	
BVS	Jump if V=1	—	REL	
CLC	0 C	C=0	IMP	
CLD	0 D	D=0	IMP	
CLI	0 I	I=0	IMP	
CLV	0 V	V=0	IMP	
CMP	A – M	N, Z, C	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
CPX	X – M	N, Z, C	IMM, ZP, ABS	
CPY	Y – M	N, Z, C	IMM, ZP, ABS	
DEC	M-1 M	N, Z	ACC,ZP, ZPX, ABS, ABSX	
DEX	X-1 X	N, Z	IMP	
DEY	Y-1 Y	N, Z	IMP	
EOR	A.XOR. M A	N, Z	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
INC	M+1 M	N, Z	ACC,ZP, ZPX, ABS, ABSX	

INX	X+1	X	N, Z	IMP	
INY	Y+1	Y	N, Z	IMP	
JMP	(PC+1) (PC+2)	PCL, PCH	—	ABS, IND, INDX	
JSR	(PC+1) (PC+2) PC+2	PCL, PCH STACK	—	ABS	
LDA	M	A	N, Z	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
LDX	M	X	N, Z	IMM, ZP, ZPY, ABS, ABSY	
LDY	M	Y	N, Z	IMM, ZP, ZPX, ABS, ABSX	
LSR	0	b7 --- b0	A	N=0, Z, C	ACC, ZP, ZPX, ABS, ABSX
NOP	—		—	IMP	
ORA	A.OR.	M A	N, Z	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
PHA	A	STACK	—	IMP	
PHP	P (status)	STACK	—	IMP	
PLA	STACK	A	N, Z	IMP	
PLP	STACK	P (status)	ALL	IMP	
ROL	C	ACC or M	C	N, Z, C	ACC, ZP, ZPX, ABS, ABSX
ROR	C	ACC or M	C	N, Z, C	ACC, ZP, ZPX, ABS, ABSX
RTI	STACK STACK-1 STACK-2	P PCH PCL	ALL	IMP	
RTS	STACK PC+1	PC PC	—	IMP	
SBC	A-M- C	A	N, Z, C, V	IMM, ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
SEC	1	C	C=1	IMP	
SED	1	D	D=1	IMP	
SEI	1	I	I=1	IMP	
STA	A	M	—	ZP, ZPX, ABS, ABSX, ABSY, INDX, INDY	
STX	X	M	—	ZP, ZPY, ABS	
STY	Y	M	—	ZP, ZPX, ABS	
TAX	A	X	N, Z	IMP	
TAY	A	Y	N, Z	IMP	
TSX	S	X	N, Z	IMP	
TXA	X	A	N, Z	IMP	
TXS	X	S	—	IMP	
TYA	Y	A	N, Z	IMP	

Note :

B: Break flag
 N: Negative flag
 Z: Zero Flag
 C: Carry flag
 I: Interrupt disable flag
 D: Decimal flag
 V: Overflow flag
 A (ACC): Accumulator
 X: Index X register
 Y: Index Y register
 S: Stack pointer
 M: Memo

Address Mode:

ACC: Operation in ACC; A

IMM: Immediate data; #dd

ZP: Zero page address; aa

ZPX: Zero page address with index X; aa, X

ZPY: Zero page address with index Y; aa, Y

ABS: Absolute address; aaaa

ABSX: Absolute address with index X; aaaa, X

ABSY: Absolute address with index Y; aaaa, Y

REL: Relative address; aa

IMP: Implied address;

Note :

#dd: A 8-bit data, the range can be #0 — #255

aa: An 8-bit address; the range can be \$00 — \$FF

aaaa: A 16-bit address; the range can be \$0000 — \$FFFF

• JA23XXX series Macro Instruction definition

a. Macro Library (Sub-program)

MAPPING.ASM (LMAPPING.ASM)

Description :

Set the special register or RAM location.

INTREQUEST.ASM (L INTREQUEST.ASM)

Description :

Set the AUD output initialization and start address, voice length, sample rate, and enable interrupt function and Implement the voice synthesis for ADPCM or PCM format

Note :

1. The macro files of MAPPING, INTREQUEST are used for JA23384 – JA231K0 bodies.
2. The macro files of LMAPPING, LINTREQUEST are used for JA23048 & JA23060.

User Program Definition

Data RAM & Register Definition (mapping.asm)

```
...
DAC0BUF equ $E0
VOICE equ DAC0BUF+1 ($E1)
CODETEMP equ VOICE+1 ($E2)
LENGL equ CODETEMP+1 ($E3)
LENGH equ LENGL+1 ($E4)
GNBUF equ LENGH+1 ($E5)
BUFFER equ GNBUF+1 ($E6)
ADDRL equ BUFFER+1 ($E7)
ADDRM equ ADDRL+1 ($E8)
ADDRH equ ADDR+1 ($E9)
MAFLAG equ ADDR+1 ($EA)
SYSCLOCK equ MAFLAG+1 ($EB)
REPEAT equ SYSCLOCK+1 ($EC)
...
```

Note :

1. User should reserve \$FF -- \$F8 for interrupt stack at least.
2. The RAM address of \$E0 -- \$EC is reserved for the sub-program of voice synthesizer.

MAFLAG bit Definition

End/Normal				Stop/Play	SR		PWM/DAC
------------	--	--	--	-----------	----	--	---------

- a. B7 (END/Normal): When voice is end, B7 (End/Normal) will be set to "1", user can check this bit to know if voice is end or not.
- b. B3 (Stop/Play): Stop the voice (1) or Start the voice (0)
- c. B2 (SR): To set the sample rate is User defined (1) or Default (0)
- d. B0 (PWM/DAC): To define the audio output is PWM (1) or DAC (0) type

Sample Rate Table

SR	CODE	SR	CODE	SR	CODE
3.0K	AC	5.2K	CF	9.5K	E5
3.1K	AF	5.3K	D0	9.8K	E6
3.2K	B1	5.4K	D1	10.2K	E7
3.3K	B4	5.6K	D3	10.7K	E8
3.4K	B6	5.7K	D4	11.1K	E9
3.5K	B8	5.8K	D4	11.6K	EA
3.6K	BA	6.0K	D6	12.2K	EB
3.7K	BC	6.1K	D7	12.8K	EC
3.8K	BE	6.2K	D7	13.5K	ED
3.9K	BF	6.4K	D8	14.2K	EE
4.0K	C1	6.6K	DA	15.1K	EF
4.1K	C3	6.7K	DA	16.0K	F0
4.2K	C4	6.9K	DB	17.1K	F1
4.3K	C5	7.1K	DC	18.3K	F2
4.4K	C7	7.3K	DD	19.7K	F3
4.5K	C8	7.5K	DE	21.3K	F4
4.6K	C9	7.8K	DF	23.3K	F5
4.7K	CA	8.0K	E0	25.6K	F6
4.8K	CB	8.3K	E1	28.4K	F7
4.9K	CC	8.5K	E2	32.0K	F8
5.0K	CE	8.8K	E3		
5.1K	CE	9.1K	E4		

Program Example

1. Special register and data RAM setting

```
.include mapping.asm (User RAM definition)
```

```
...
```

2. Program start setting and include sub-program

```
org $xxxx ; User program start
```

```
START:
```

```
.include INTREQUEST.asm; Include the voice processing sub-program
```

```
...
```

```
...
```

3. Voice play setting for user program

```
*****
```

```
; This program is used to call the voice synthesizer
```

```
lda #$aa ; Set the volume
```

```
sta VOLC
```

```
lda #$00
```

```
sta VOICE ; Initialize the VOICE flag
```

```
lda #$02 ; Start play bit, default sample rate, mono & DAC output
```

```
sta MAFLAG ; Set the MAFLAG register
```

```
lda #$XX ; Look up sample rate table to set sample
```

```
sta TMR1L ; If have set SR flag in MAFLAG
```

```
lda #$XX
```

```
sta ADDR1
```

```
lda #$XX
```

```
sta ADDR2
```

```
lda #$XX ; The start address can get from *.idx file
```

```
sta ADDR3 ; To set the voice start address,
```

```
lda #$nn
```

```
sta REPEAT
```

```
jsr INITIAL ; To call the voice initial sub-program
```

```
; If JA23048 & JA23060, "INITIAL" will be replaced by "LINITIAL"
```

```
jsr PLAY ; To call the voice play sub-program
```

```
; If JA23048 & JA23060, "PLAY" will be replaced by "LPLAY"
```

```
*****
```

```
;
```

```
...
```

```
...
```

4. Check the voice play is terminated or not

```
; User to confirm the voice whether is terminated or not?
```

```
CheckVoice:
```

```

lda    $80
bit    MAFLAG
beq    CheckVoice    ; If voice is playing
jmp    OtherProgram  ; If voice is end
...

```

5. Interrupt request declaration

INTREQUEST:

```

pha
txa
pha
lda    #$01
bit    INTF
...
lda    #$02
bit    INTF
beq    OtherINT
jsr    VOICEINT    ; Voice synthesizer sub-program label

```

; If using JA23048 or JA23060, "VOICEINT" should be replaced by "LVOICEINT"

OtherINT:

```

...
pla
tax
pla
cli
rti

```

6. Include the voice synthesizer

```
.include INTREQUEST.ASM
```

7. Power-on and Interrupt vector declaration

```

org    $ffc
dw    START    ; Power-on or reset start vector
dw    INTREQUEST
end

```

Revision History

Date	Version	Comment
2005.4.28	0.7	1. Add AD5 format. P18 2. Modify Data format. P18