MLC0xxB Series

8-bit I/O type micro-controller with voice function

Features

- Single chip 8-bit CPU
- Operation voltage: 2.4V to 5.5V
- Memory:
  - ROM (shared by program and data): 1024K ~ 48KBytes
  - Data RAM: 256 Bytes
- 24 input/output pins with wake-up function
- Two power-down modes for saving power consumption:
  - Sleep mode: micro-controller no operation (main- and sub-oscillator still oscillating)
  - Stop mode: micro-controller no operation (all oscillators stop oscillating)
- Two current DAC output for voice synthesizer
- Dual-channel melody with programmable envelope
- Programmable sample rate for voice/melody function
- One serial input port and voltage comparator built-in
- Three re-loadable 16-bit timers
- One watchdog timer built-in
- Oscillator
  - Single or dual clock operation is selected by code option
  - Main oscillator operation at crystal or RC mode is selected by code option
  - Crystal/Ceramic oscillator up to 4MHz @ 2.4V and 8MHz @ 3.6V
  - RC oscillator up to 4MHz @ 2.4V

Selection Information

		MLC331B	MLC241B	MLS161B	MLC121B
ROM (Program ROM)		1024K x 8-bit (32K x 8-bit)	768K x 8-bit (32K x 8-bit)	512K x 8-bit (32K x 8-bit)	384K x 8-bit (32K x 8-bit)
I/O		24	24	24	24
Voice Duration	6KHz 4-bit ADPCM	340 sec	250 sec	165 sec	125 sec
	8KHz 4-bit ADPCM	250 sec	190 sec	125 sec	90 sec

		MLC081B	MLC061B	MLC041B	MLC031B	MLC021B	MLC017B
ROM (Program ROM)		256K x 8-bit (32K x 8-bit)	192K x 8-bit (32K x 8-bit)	128K x 8-bit (32K x 8-bit)	96K x 8-bit (32K x 8-bit)	64K x 8-bit (32K x 8-bit)	48K x 8-bit (32K x 8-bit)
I/O		24	24	24	24	24	24
Voice Duration	6KHz 4-bit ADPCM	80 sec	60 sec	40 sec	30 sec	20 sec	16 sec
	8KHz 4-bit ADPCM	60 sec	45 sec	30 sec	22 sec	16 sec	12 sec

MEGAWIN

Application Field

General voice synthesizer

Toy controller

General IR controller

General Description

MLC0xxB series integrates an 8-bit CPU core, SRAM, timer, D/A and system control circuits by a CMOS silicon gate technology. The ROM can store voice, melody, data table and program.

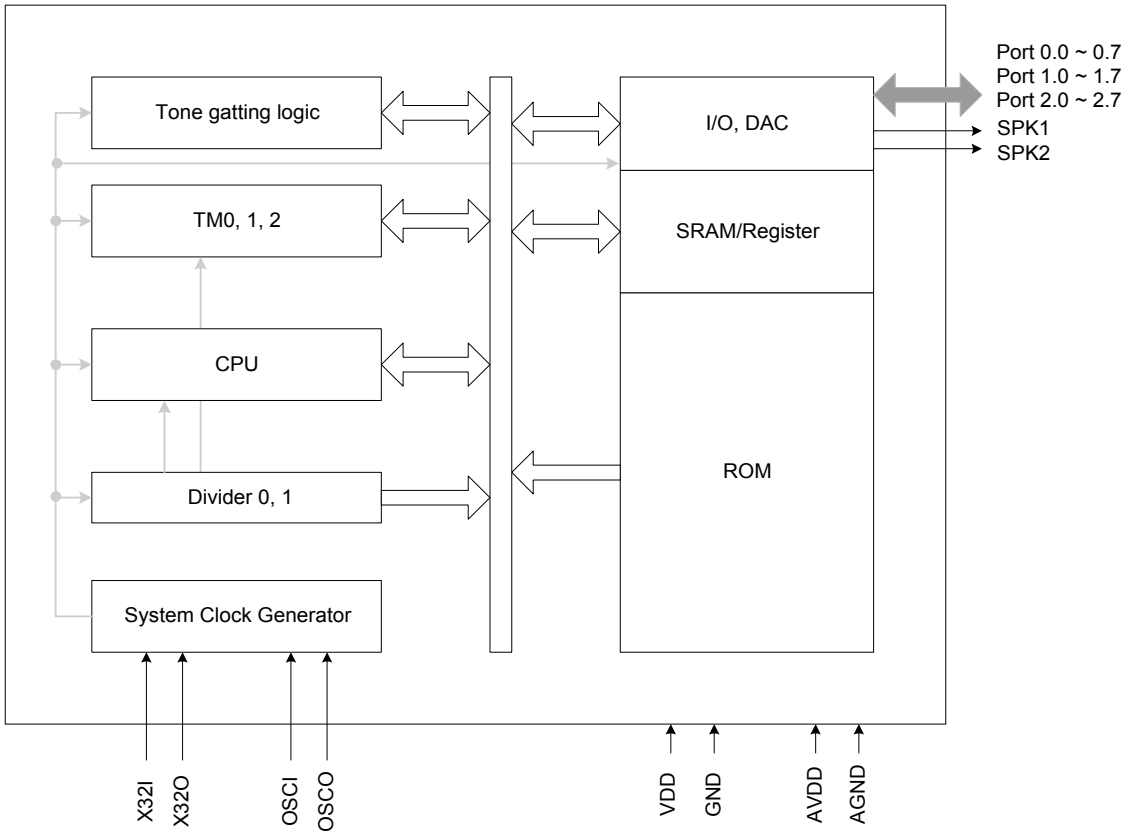
Twenty-four I/O pins can be used for keypad control, motor control, IR application, LED indicators or communication with other systems. This chip can implement a dual tone melody function with programmable envelope, which can perform harmonic music with different timbres.

This chip is very suitable for instruments, speech products, and intelligent educational toys, etc.

Pad Description

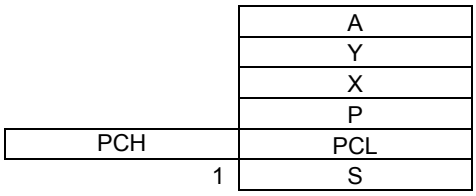
Pad No.	Pad Name	I/O	Description
33, 4	AGND, GND	P	Ground pins, the two ground pins should be connected at the outside individually
35, 1	AV_{DD}, V_{DD}	P	Positive power pins, the two power pins should be connected at the outside individually
2, 3	OSCO, OSCI	O, I	RC or crystal oscillator pins
7, 8	X32O, X32I	O, I	32.768KHz crystal oscillator pins
5	/RES	I	System reset pin (low active)
6	TEST	-	For test mode only
36	SPK2	O	DAC 2 output
34	SPK1	O	DAC 1 output
9 ~ 16	P0.0 ~ P0.7	I/O	Programmable I/O ports with interrupt function
25 ~ 32	P1.0 ~ P1.7	I/O	Programmable I/O ports. Port P1.3, P1.4, P1.5 can be I/O or serial input port. Port1.6, 1.7 can be output with IR carrier.
17 ~ 24	P2.0 ~ P2.7	I/O	Programmable I/O ports. Port 2.4~2.6 can be I/O or voltage comparator.

Block Diagram



Function Description

Registers



Accumulator

The accumulator is a general-purpose 8-bit register, which stores the results of most arithmetic and logic operations. In addition, the accumulator usually contains one of two data words, which are used in these operations.

Index Register (X, Y)

There are two 8-bit index registers (X and Y), which may be used to count program steps or to provide an index value to be used in generating an effective address. When executing an instruction, which specifies indexed addressing, the CPU fetches the OP Code and the base address, and modifies the address by adding the index register to it prior to performing the desired operation. Pre- or post-index of index address is possible.

Processor Status Register (P)

The 8-bit processor status register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both the program and the CPU.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N	V	1	B	D	I	Z	C

N: Signed flag, 1 = negative, 0 = positive

V: Overflow flag, 1 = true, 0 = false

B: BRK interrupt command, 1 = BRK, 0 = IRQB

D: Decimal mode, 1 = true, 0 = false

I: IRQB disable flag, 1 = disable, 0 = enable

Z: Zero flag, 1 = true, 0 = false

C: Carry flag, 1 = true, 0 = false

Program Counter (PC)

The 16-bit program counter register provides the addresses, which step the micro-controller through sequential program instructions. Each time the micro-controller fetches an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order 8 bits of the address bus and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time an instruction or data is fetched from program memory.

Stack Pointer (S)

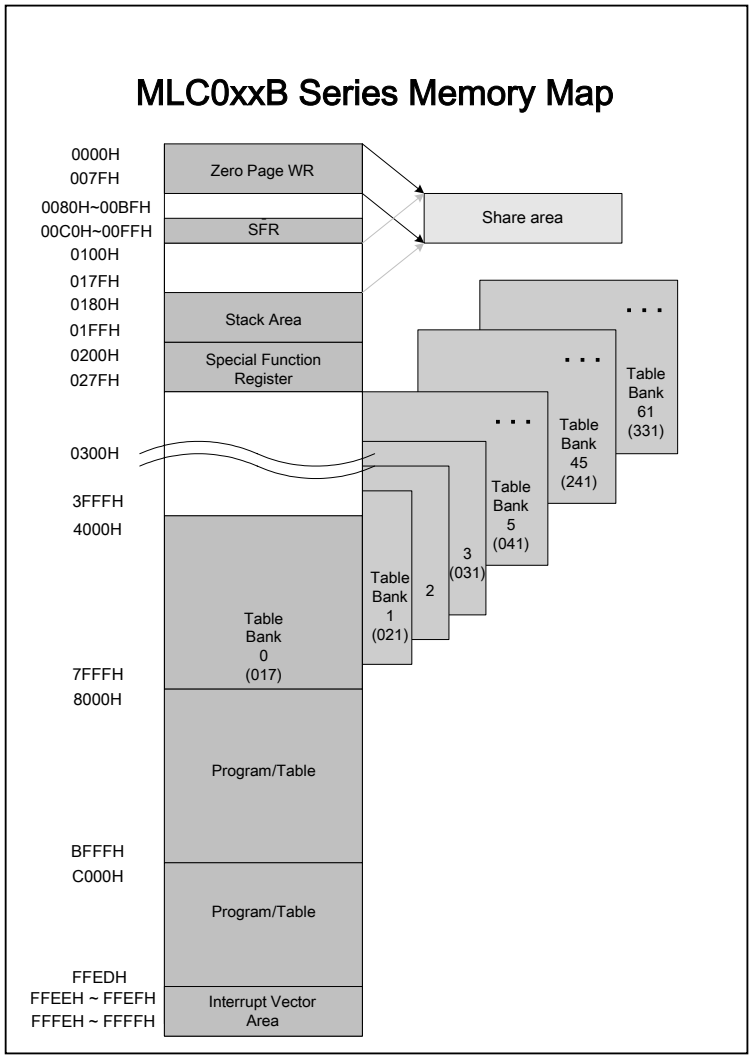
The stack pointer is an 8-bit register, which is used to control the addressing of the variable-length stack. The stack pointer is automatically incremented and decremented under control of the micro-controller to perform stack manipulations under direction of either the program or interrupts (/NMI or /IRQ). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by the user's software.

Memory Map

There are 256 bytes SRAM in MLC0xxB series. They are working RAM (0000H to 007FH) and stacks (0180H to 01FFH). The address 0100H to 17FH are shared with address 0000H to 007FH. The address 00C0H to 00FFH and 0200H to 027FH are special function registers area.

The bank select function, ranged from 4000H to 7FFFH, is used for extending memories if the ROM size is more than 32K bytes in MLC0xxB series. The default bank number is 00H after power on or reset.

There are 1024K ~ 48K bytes program/data ROM in MLC0xxB series. It is combined with 32K program/data ROM and bank switching data ROM. The ROM address from 4000H to FFFFH can store program, voice data, melody notes and other data. The address mapping of MLC0xxB series is shown as below.



Special Function Register (SFR)

The address 00C0H to 00FFH and 0200H to 027FH are reserved for special function registers (SFR). The SFR is used to control or store the status of I/O, timers, system clock and other peripheral.

SFR (special function register): 00C0H ~ 00FFH (page 0 area)

Address	Content	Default	Address	Content	Default
00C0	NMI_SEL	00	00D0	BANK	00
00C1		X	00D1		X
00C2	IRQ_EN / IRQ_ST	00	00D2		X
00C3	IRQ_CLR	00	00D3		X
00C4	TM0L	00	00D4		X
00C5	TM0H	00	00D5	P1_MFR	00
00C6	TM0_CTL	00	00D6	P2_MFR	00
00C7	TM0_MOD	00	00D7		X
00C8	TM1L	00	00D8	P0	00
00C9	TM1H	00	00D9	P1	00
00CA	TM1_CTL	00	00DA	P2	00
00CB		X	00DB		X
00CC	DIV0_ST / DIV0x_SEL	00	00DC		X
00CD		X	00DD		X
00CE	DIV1_STL/ DIV1x_SEL	00	00DE	WDT_CTL	00
00CF	DIV1_STH	00	00DF	WDT_CLR	00

Address	Content	Default	Address	Content	Default
00E0		X	00F0		X
00E1	CH1	00	00F1		X
00E2		X	00F2		X
00E3	CH2	00	00F3		X
00E4		X	00F4		X
00E5	CH3	00	00F5		X
00E6		X	00F6		X
00E7		X	00F7		X
00E8	TM2_L	00	00F8	CMP_CTL	00
00E9	TM2_H	00	00F9	DB_TC	00
00EA	TM2_CTL	00	00FA	VT_CTL	00
00EB		X	00FB		X
00EC		X	00FC		X
00ED		X	00FD	DAC_DRV	00
00EE		X	00FE		X
00EF		X	00FF		X

SFR (special function register): 0200H ~ 027FH

Address	Content	Default	Address	Content	Default
0200	PWR_CR	00	0210		X
0201	FCPU_SR	00	0211		X
0202	RLH_EN	00	0212		X
0203		X	0213		X
0204		X	0214		X
0205		X	0215		X
0206		X	0216		X
0207		X	0217		X
0208		X	0218		X
0209		X	0219		X
020A		X	021A		X
020B		X	021B		X
020C		X	021C		X
020D		X	021D		X
020E		X	021E		X
020F		X	021F		X

Address	Content	Default	Address	Content	Default
0220		X	0230		X
0221		X	0231		X
0222		X	0232		X
0223		X	0233		X
0224		X	0234		X
0225		X	0235		X
0226		X	0236		X
0227		X	0237		X
0228		X	0238		X
0229		X	0239		X
022A		X	023A		X
022B		X	023B		X
022C		X	023C		X
022D		X	023D		X
022E		X	023E		X
022F		X	023F		X

Special Function Register, Continued

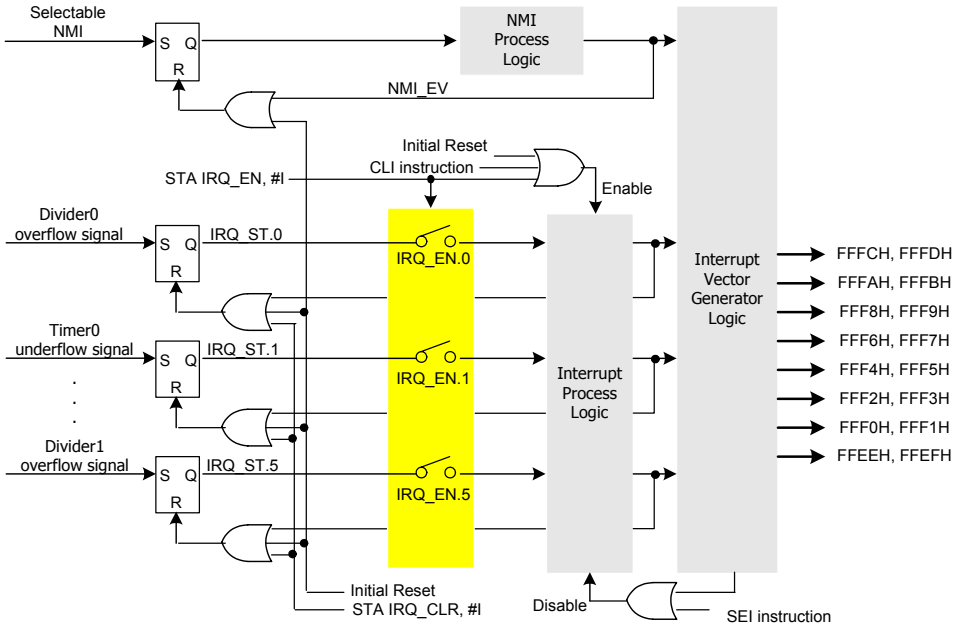
Address	Content	Default	Address	Content	Default
0240	P0CR	00	0250		X
0241	P0MR	00	0251		X
0242		X	0252		X
0243		X	0253		X
0244	P1CR	00	0254		X
0245	P1MR	00	0255		X
0246		X	0256		X
0247		X	0257		X
0248	P2CR	00	0258		X
0249	P2MR	00	0259		X
024A		X	025A		X
024B		X	025B		X
024C		X	025C		X
024D		X	025D		X
024E		X	025E		X
024F		X	025F		X

Address	Content	Default	Address	Content	Default
0260		X	0270		X
0261		X	0271		X
0262		X	0272		X
0263		X	0273		X
0264		X	0274		X
0265		X	0275		X
0266		X	0276		X
0267		X	0277		X
0268		X	0278		X
0269		X	0279		X
026A		X	027A		X
026B		X	027B		X
026C		X	027C		X
026D		X	027D		X
026E		X	027E		X
026F		X	027F		X

Interrupt Vectors

Vector Address	Item	Priority	Properties	Memo
FFFCH, FFFDH	RESET	1	Ext.	Initial reset
FFFAH, FFFBH	NMI	2	Int./Ext.	Non-maskable interrupt vector
FFF8H, FFF9H	DIV0x	3	Int.	Selectable divider 0 carry out interrupt
FFF6H, FFF7H	TM0	4	Int.	Timer 0 overflow interrupt
FFF4H, FFF5H	P0	5	Ext.	Port P0 interrupt vector
FFF2H, FFF3H	TM1	6	Int.	Timer 1 overflow interrupt
FFF0H, FFF1H	TM2	7	Int.	Timer 2 overflow interrupt
FFEEH, FFEFH	DIV1x	8	Int.	Selectable divider 1 carry out interrupt

There are eight kinds of interrupt sources are provided in MLC0xxB series. The flag IRQ_EN and IRQ_ST are used to control the interrupts. When flag IRQ_ST is set to '1' by hardware and the corresponding bits of flag IRQ_EN has been set by software, an interrupt is generated. When an interrupt occurs, all of the interrupts are inhibited until the CLI or STA IRQ_EN, #I instruction is invoked. Executing the SEI instruction can also disable the interrupts.



Interrupt Registers

NMI select flag

Address	Register	7	6	5	4	3	2	1	0	R	W
00C0H	NMI_SEL	-	-	-	-	-	NIS2	NIS1	NIS0	√	√

NIS2	NIS1	NIS0	Selected NMI source
0	0	0	None (default)
0	0	1	TM0
0	1	0	DIV0x
0	1	1	P0
1	0	0	TM1
1	0	1	TM2
1	1	0	DIV1x
1	1	1	(None)

This register is used to select the NMI trigger source. The NMI is a rare resource of this system. Only one trigger source is selected at one application is recommended. If over one trigger source is needed in some special applications, program must to distinguish the additional interrupter. After NMI occurs, program has to read NMI_SEL register to know which source triggering NMI.

IRQ enable flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C2H	IRQ_EN	-	-	DIV1x	TM2	TM1	P0	TM0	DIV0x	-	√

Program can enable or disable the ability of triggering IRQ through this register.

0: Disable (default “0” at initialization)

1: Enable

P0: Raising or falling edge occurs at port 0 input mode

TM0, TM1, TM2: Timer 0/1/2 underflow

DIV0x, DIV1x: Divider 0/1 selected interrupt frequency occurred

IRQ status flag (same address with IRQ_EN)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C2H	IRQ_ST	-	-	DIV1x	TM2	TM1	P0	TM0	DIV0x	√	-

When IRQ occurs, program can read this register to know which source triggering IRQ.

IRQ clear flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C3H	IRQ_CLR	-	-	DIV1x	TM2	TM1	P0	TM0	DIV0x	-	√

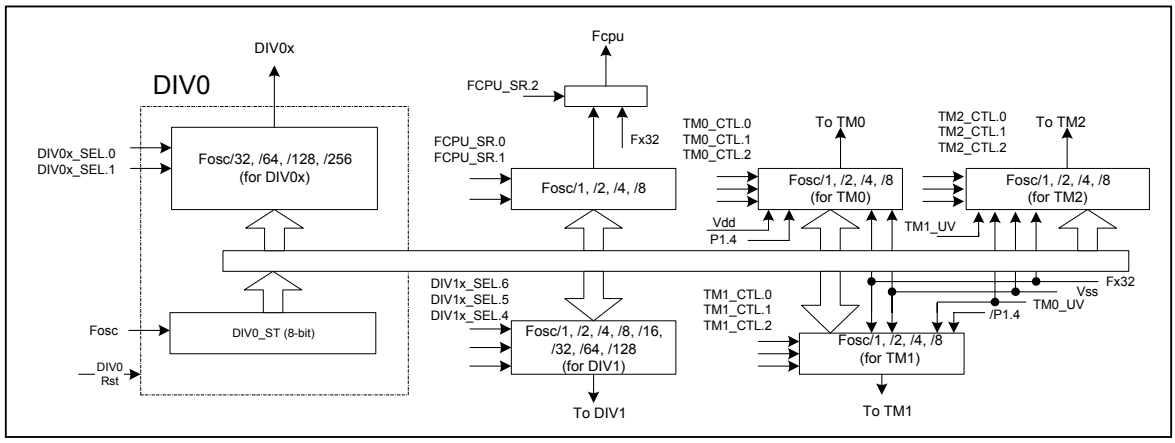
Program can clear the interrupt event by writing ‘1’ into the corresponding bit.

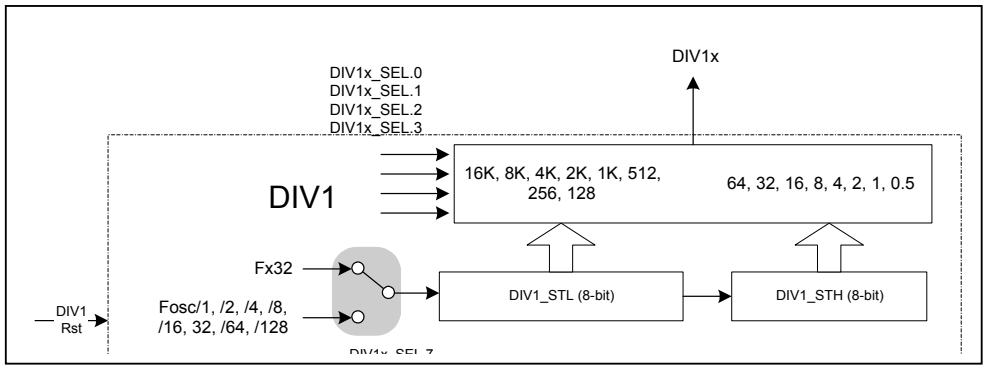
DIV0 interrupt selector (clock source: Fosc)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00CCH	DIV0_ST	Fosc /256	Fosc /128	Fosc /64	Fosc /32	Fosc /16	Fosc /8	Fosc /4	Fosc /2	√	-
00CCH	DIV0x_SEL	-	-	-	-	-	-	CKO1	CKO0	-	√

The divider 0 is organized as an 8-bit binary up counter, which is designed to generate periodic interrupts. When the main oscillator starts action, the divider 0 is incremented by each clock (Fosc). The contents of divider 0 can be reset to 00H by POR, reset, waken from STOP mode and change the contents of DIV0x_SEL.

CKO1	CKO0	Selected DIV0x frequency
0	0	Fosc / 32
0	1	Fosc / 64
1	0	Fosc / 128
1	1	Fosc / 256





DIV1 interrupt selector (If the frequency of divider 1 clock source is 32.768KHz)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00CEH	DIV1_STL	128 Hz	256 Hz	512 Hz	1024 Hz	2048 Hz	4096 Hz	8192 Hz	16384 Hz	√	-
00CFH	DIV1_STH	0.5Hz	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	√	-
00CEH	DIV1x_SEL	CKI7	CKI6	CKI5	CKI4	CKO3	CKO2	CKO1	CKO0	-	√

The divider 1 contents can be reset to 00H by POR, reset, waken from STOP mode and writing DIV1x_SEL register any value.

CKI7: select the input clock source of divider 1. 0: $F_{OSC/x}$ (00CE H), 1: F_{X32}

CKI6	CKI5	CKI4	Selected DIV1 input frequency
0	0	0	F_{OSC} / 1
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	F_{OSC} / 16
1	0	1	F_{OSC} / 32
1	1	0	F_{OSC} / 64
1	1	1	F_{OSC} / 128

CKO3	CKO2	CKO1	CKO0	Selected DIV1x frequency
0	0	0	0	F_{DIV1} / 2 (16384 Hz)
0	0	0	1	F_{DIV1} / 4 (8192 Hz)
0	0	1	0	F_{DIV1} / 8 (4096 Hz)
0	0	1	1	F_{DIV1} / 16 (2048 Hz)
0	1	0	0	F_{DIV1} / 32 (1024 Hz)
0	1	0	1	F_{DIV1} / 64 (512 Hz)
0	1	1	0	F_{DIV1} / 128 (256 Hz)
0	1	1	1	F_{DIV1} / 256 (128 Hz)
1	0	0	0	F_{DIV1} / 512 (64 Hz)
1	0	0	1	F_{DIV1} / 1024 (32 Hz)
1	0	1	0	F_{DIV1} / 2048 (16 Hz)
1	0	1	1	F_{DIV1} / 4096 (8 Hz)
1	1	0	0	F_{DIV1} / 8192 (4 Hz)
1	1	0	1	F_{DIV1} / 16384 (2 Hz)
1	1	1	0	F_{DIV1} / 32768 (1 Hz)
1	1	1	1	F_{DIV1} / 65536 (0.5 Hz)

Watchdog Timer (WDT)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00DEH	WDT_CTL	RSTS	-	-	-	-	RSEL	CKI1	CKI0	√	√
00DFH	WDT_CLR	CLR	-	-	-	Bit 3	Bit 2	Bit 1	Bit 0	√	√

RSTS: WDT reset status, set by hardware when WDT overflows, clear by hardware reset or set WDT_CLR.7 to one to clear this bit (this bit is read only)

RSEL: WDT reset selector, = 0 Reset whole chip except RSTS (WDT_CTL.7)

= 1 Reset PC and IRQ_EN only

CKI1, CKI0: WDT clock selector, = 00 F_{DIV1}/16384 selected (8 S @ F_{DIV1} = 32K)

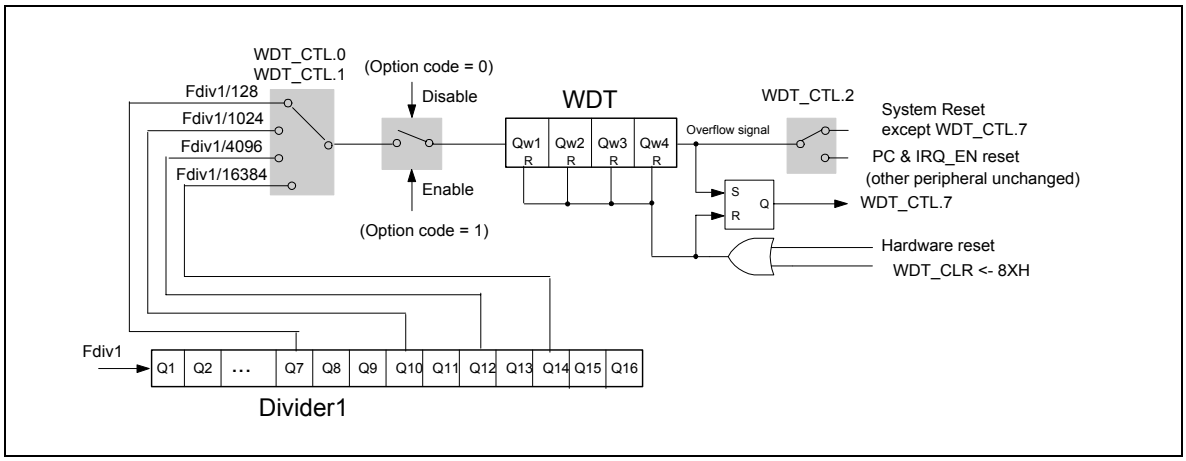
= 01 F_{DIV1}/4096 selected (2 S @ F_{DIV1} = 32K)

= 10 F_{DIV1}/1024 selected (0.5 S @ F_{DIV1} = 32K)

= 11 F_{DIV1}/128 selected (62.5 mS @ F_{DIV1} = 32K)

CLR: RSTS clear control bit, program can clear RSTS by program "1" into this bit (this bit is write only)

The watchdog timer (WDT), which is organized as a 4-bit counter, is designed to prevent the program from unknown errors. The WDT is enabling by code option. If the WDT overflows, the WDT reset function will be performed. The watchdog timer control register (WDT_CTL) controls the WDT reset function. RSTS (WDT_CTL.7) is set by hardware when the WDT overflows and is cleared by store one to the bit 7 of WDT_CLR register or hardware reset. There are two types of WDT reset, which is selected by RSEL (bit2 of WDT_CTL). WDT overflow will cause two types reset depending on the setting of RSEL — if RSEL is equal to 0, the reset is the same as hardware reset except the setting of WDT_CTL and WDT_CLR; If RSEL is equal to 1, the reset only acts on program counter (PC) and IRQ_EN. The WDT clock frequency is decided by bit1 and bit0 of WDT_CTL register. Store one to the bit 7 of WDT_CLR register will also reset the contents of the WDT. In normal operation, the application program must reset WDT before it overflows. The organization of the divider1 and watchdog timer is shown as below.



System Control Registers

Bank select

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D0H	BANK	-	BK6	BK5	BK4	BK3	BK2	BK1	BK0	√	√

Program can switch the memory bank through this register. After power on reset, this register in initialized as 00H. The maximum bank numbers in MLC0xxB series are show as below:

Part No.	MLC331B	MLC241B	MLC161B	MLC121B
Max. Bank	11 1101b	10 1101b	01 1101b	01 0101b

Part No.	MLC081B	MLC061B	MLC041B	MLC031B	MLC021B	MLC017B
Max. Bank	00 1101b	00 1001b	00 0101b	00 0011b	00 0001b	00 0000b

For more detailed information, please refer to memory map description.

Power saving control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0200H	PWR_CR	-	-	-	-	-	CKC1	CKC0	HALT	-	√

CKC1	CKC0	System clock control
0	0	F_{OSC} enable, F_{X32} enable (Dual mode)
0	1	F_{OSC} enable, F_{X32} disable (Single mode)
1	0	F_{OSC} disable, F_{X32} enable (Slow mode)
1	1	F_{OSC} disable, F_{X32} disable (Stop mode)

Note: PWR_CR.CKC0 is inhibited when single clock mode is selected.

HALT: F_{CPU} off-line control bit. 1: F_{CPU} off-line, 0: F_{CPU} on-line

Program can switch the normal operation mode to the power-saving mode for saving power consumption through this register. There are three power saving modes in this system.

Slow mode: *(PWR_CR.CKC1 = 1, PWR_CR.CKC0 = 0)*

The main uC clock (F_{OSC}) stops oscillating. Only very low power is needed for uC to keep running.

Stop mode: *(PWR_CR.CKC1 = 1, PWR_CR.CKC0 = 1)*

All system clocks stop oscillating. The uC can be awakened from stop mode by 3-ways: port 0 interrupt, hardware reset, or power-on reset.

Halt mode: *(PWR_CR.HALT = 1)*

The F_{CPU} clock in off-line status. The oscillator(s) still oscillating if the PWR_CR.CKC1, PWR_CR.CKC0 keep low. The uC can be awakened from halt mode by 3-ways: all interrupt events (DIV0x, DIV1x, timer 0, timer 1, timer 2, port 0), hardware reset, or power-on reset.

FCPU selector

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0201H	FCPU_SR	-	-	-	-	-	CKS2	CKS1	CKS0	-	√

CKS2: FCPU clock source select. 0: FOSC/x (0201 H), 1:FX32

CKS1	CKS0	Selected FOSC/x (0201H) frequency
0	0	FOSC / 1 (default)
0	1	FOSC / 2
1	0	FOSC / 4
1	1	FOSC / 8

Release halt mode enable flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0202H	RLH_EN	-	-	DIV1x	TM2	TM1	P0	TM0	DIV0x	-	√

Set IRQ_CLR register to clear the halt release event.

Release halt status flag is the IRQ_ST register.

Timers/Counters

Timer0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C4H	TM0L	T7	T6	T5	T4	T3	T2	T1	T0	√	√
00C5H	TM0H	T15	T14	T13	T12	T11	T10	T9	T8	√	√
00C6H	TM0_CTL	STC	RL/S	TKES	TMS1	TMS0	TKI2	TKI1	TKI0	√	√
00C7H	TM0_MOD	-	-	-	TKPS	SRS	-	TDI1	TDI0	√	√

STC: Start/Stop counting. 1: start and pre-load the value to counter, 0: stop timer clock (set this bit to 1 will be ignored when this bit already set to 1)

RL/S: Auto-reload disable/enable. 1: disable auto-reload, 0: enable auto-reload

TKES: Event or series input clock-in trigger edge selector; 0: rising edge, 1: falling edge

TMS1	TMS0	Select TM0 operation mode
0	0	16-bit counter (default)
0	1	Reserved
1	0	16-bit shift register (the $F_{TM0\_UV}/2$ circuit will be bypassed)
1	1	16-bit rotate register (the $F_{TM0\_UV}/2$ circuit will be bypassed)

TKI2	TKI1	TKI0	Selected TM0 input clock source
0	0	0	F_{OSC} / 1
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	V_{DD}
1	0	1	F_{X32}
1	1	0	P1.4
1	1	1	V_{SS}

TKPS: Exchange the clock and data path of timer 0. 0:default path, 1: exchanged path

SRS: Shift register selector. 0: shift left, 1: shift right

TDI1	TDI0	Selected TM0 shift-in data source
0	0	V_{DD}
0	1	P1.5
1	0	DIV0x
1	1	DIV1x

Timer1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C8H	TM1L	T7	T6	T5	T4	T3	T2	T1	T0	√	√
00C9H	TM1H	T15	T14	T13	T12	T11	T10	T9	T8	√	√
00CAH	TM1_CTL	STC	RL/S	TKES	TMS1	-	TKI2	TKI1	TKI0	√	√

STC: Start/Stop counting. 1: start and pre-load the value to counter, 0: stop timer clock

RL/S: Auto-reload disable/enable. 1: disable auto-reload, 0: enable auto-reload

TKES: Event or series input clock-in trigger edge selector; 0: rising edge, 1: falling edge

TMS1	Select TM1 operation mode
0	16-bit counter
1	16-bit shift right register (the F_{TM1_UV}/2 circuit will be bypassed)

TKI2	TKI1	TKI0	Selected TM1input clock source
0	0	0	F_{OSC} / 1
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	TM0 underflow
1	0	1	F_{X32}
1	1	0	/P1.4
1	1	1	V_{SS}

Timer 1 is a 16-bit down-count counter. The counter underflow frequency of timer 1 can be calculated with the equation:

$$F_{TM1\_UV} = F_{TM1} / (TM1+1)$$

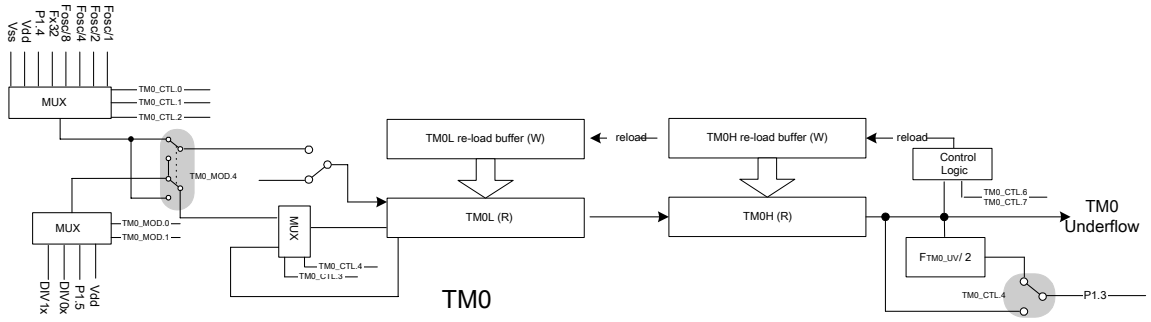
For example: (if F_{TM1} = 4.096MHz)

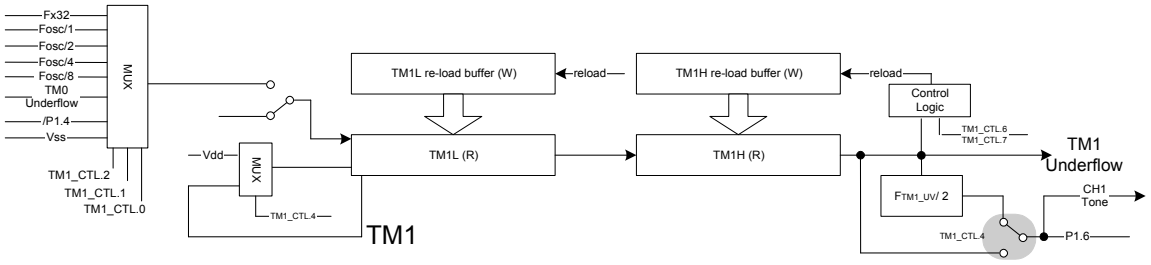
TM1	Frequency
00 00H	Invalid
00 01H	2.048MHz
00 02H	1.365MHz
…	…
00 FFH	16KHz
…	…
FF FFH	62.5Hz

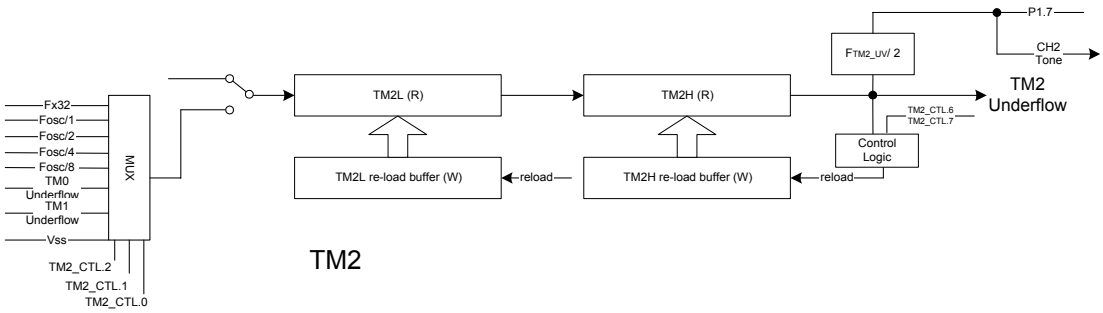
The timer 1 also can be used as tone generator. It generates specific frequency of tone with square wave, but the frequency of specific tone is half of the overflow frequency. The example frequency table is shown as below:

Set TM1_CTL to be 80H (enable counting and auto reload, source clock = 4.00MHz)

TM1H	TM1L	Underflow frequency	Tone frequency	Relative scale of tone
3BH	BAH	261.609	130.804	C3 (130.813)
38H	5EH	277.200	138.600	C3# (138.591)
35H	36H	293.643	146.821	D3 (146.832)
…	…	…	…	…
1DH	DCH	523.286	261.643	C4 (261.626)
…	…	…	…	…
0EH	EEH	1046.572	523.286	C5 (523.251)
…	…	…	…	…
07H	77H	2093.144	1046.572	C6 (1046.502)
…	…	…	…	…
03H	F4H	3952.569	1976.284	B6 (1975.533)
03H	BCH	4184.100	2092.050	C7 (2093.005)







Timer2

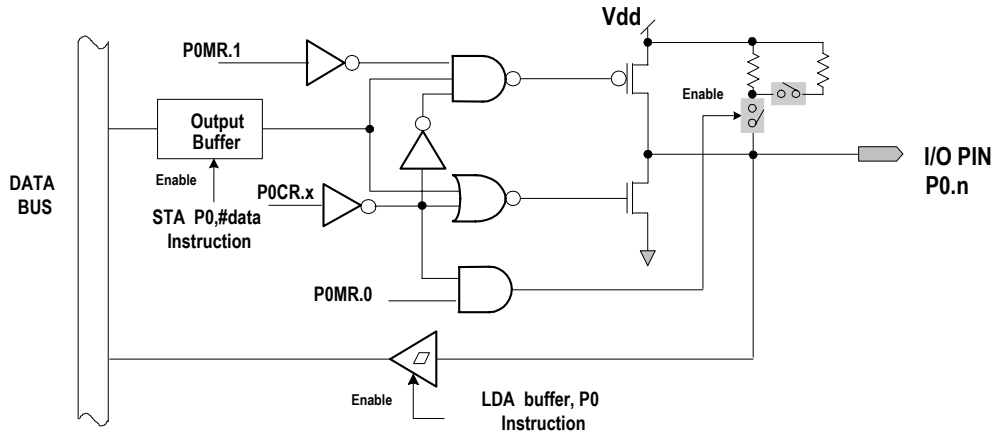
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00E8H	TM2_L	T7	T6	T5	T4	T3	T2	T1	T0	√	√
00E9H	TM2_H	T15	T14	T13	T12	T11	T10	T9	T8	√	√
00EAH	TM2_CTL	STC	RL/S	-	-	-	TKI2	TKI1	TKI0	√	√

STC: Start/Stop counting. 1: start and pre-load the value to counter, 0: stop timer clock

RL/S: Auto-reload disable/enable. 1: disable auto-reload, 0: enable auto-reload

TKI2	TKI1	TKI0	Selected TM2 input clock source
0	0	0	F_{OSC} / 1
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	TM0 underflow
1	0	1	F_{X32}
1	1	0	TM1 underflow
1	1	1	V_{SS}

Input/Output Pin of the P0



I/O Ports

Port 0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D8H	P0	P07	P06	P05	P04	P03	P02	P01	P00	√	√
0240H	P0CR	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	√	√
0241H	P0MR	-	MP06	MP05	MP04	-	MP02	MP01	MP00	√	√

Port 0 is an 8-bit I/O port; each pin can be programmed as input or output individually.

P0CR: p0.0~p0.7 is input or output. 0: input, 1: output

P0MR: p0.0~p0.7, pull-high, CMOS/NMOS and pull-high value setting

P0MR.0: P0.0 ~ P0.3 Pull-high control, 0: disable, 1:enable

P0MR.1: P0.0 ~ P0.3 CMOS/NMOS selector, 0: CMOS, 1:NMOS

P0MR.2: P0.0 ~ P0.3 Pull-high resistor value control, 0: large, 1: small

P0MR.4: P0.4 ~ P0.7 Pull-high control, 0: disable, 1: enable

P0MR.5: P0.4 ~ P0.7 CMOS/NMOS selector, 0: CMOS, 1:NMOS

P0MR.6: P0.4 ~ P0.7 Pull-high resistor value control, 0: large, 1: small (350K or 50K selector)

At initial reset, the port P0 is all in input mode. Each pin of port P0 can be specified as input or output mode independently by the P0CR registers. When P0 is used as output port, CMOS or NMOS open drain output type can be selected by the P0MR register. Port P0 has the internal pull-high resistors that can be enabled/disabled by specifying the P0MR.0 and P0MR.4 respectively. The pull-high resistors will be temporarily disable if the port is specified as output mode. The read value will be the contents of output buffer in output mode. When P0 port is used as input mode and the RLH_EN, and IRQ_EN corresponding to the P0 port are set, a signal change at the port P0 (any pin) will execute the halt mode release or interrupt subroutine. Both the raising or falling signal will set the port P0 event. The Schmitt trigger circuit is added in the input port part of all I/O pins.

Please set port 0 as output high before set it as input mode, if speeds up the internal pull-high effect is needed. If the I/O ports are not used in your application, please set them as input with pull-high or output mode to avoid unnecessary power consumption.

Port 1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D9H	P1	P17	P16	P15	P14	P13	P12	P11	P10	√	√
0244H	P1CR	CP17	CP16	CP15	CP14	CP13	CP12	CP11	CP10	√	√
0245H	P1MR	-	-	MP15	MP14	-	-	MP11	MP10	√	√

Port 1 is an 8-bit I/O port; refer to port 0 for more information.

P1CR: P1.0 ~ P1.7 is input or output. 0: input, 1: output

P1MR: P1.0 ~ P1.7, pull-high and CMOS/NMOS

Port 1 multi-function selector

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D5H	P1_MFR	PS7	PS6	-	-	PS3	-	-	-	√	√

The port 1 can be programmed to special function via P1_MFR register. The serial input port is multiplex with P1.4 and P1.5 (P1.4/CLK and P1.5/Din), the event counter input port is multiplex with P1.4 and P1.5, and the Infrared control is multiplex with P1.3, P1.6 or P1.7

PS7, PS6, PS3: Normal I/O or TM2/TM1/TM0 carrier output selector. 0:normal I/O, 1: carrier output

Port 2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00DAH	P2	P27	P26	P25	P24	P23	P22	P21	P20	√	√
0248H	P2CR	CP27	CP26	CP25	CP24	CP23	CP22	CP21	CP20	√	√
0249H	P2MR	-	-	MP25	MP24	-	-	MP21	MP20	√	√

Port 2 is an 8-bit I/O port; refer to port 0 for more information.

P2CR: P2.0 ~ P2.7 is input or output. 0: input, 1: output

P2MR: P2.0 ~ P2.7, pull-high and CMOS/NMOS

Port 2 multi-function selector (P2.4/V-, P2.5/V+, P2.6/Vo)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D6H	P2_MFR	-	PS6	-	-	-	-	-	-	√	√

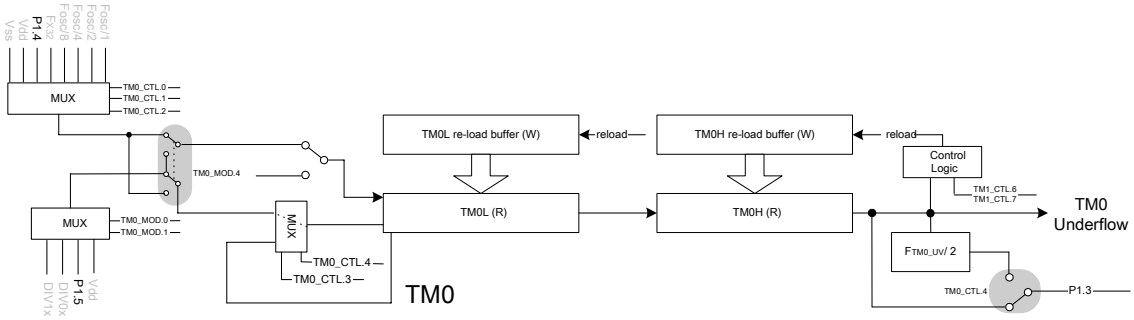
PS6: Normal I/O or voltage comparator output selector of port 2.6. 0:normal I/O, 1: comparator output

Voltage comparator (P2.4/V-, P2.5/V+, P2.6/Vo)

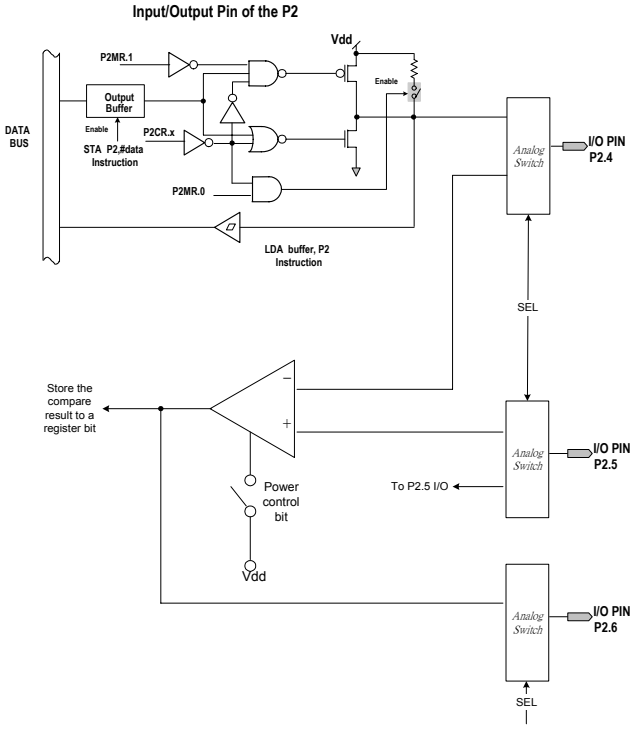
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00F8H	CMP_CTL	-	-	-	-	-	PWR	-	RLT	√	√

PWR: Voltage comparator power control. 0: power-off, 1: power-on

RLT: The voltage compare result. 0: V+ < V-, 1: V+ > V-



Use TM0 as series input buffer



CH1 Buffer

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00E1H	CH1	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	√	√

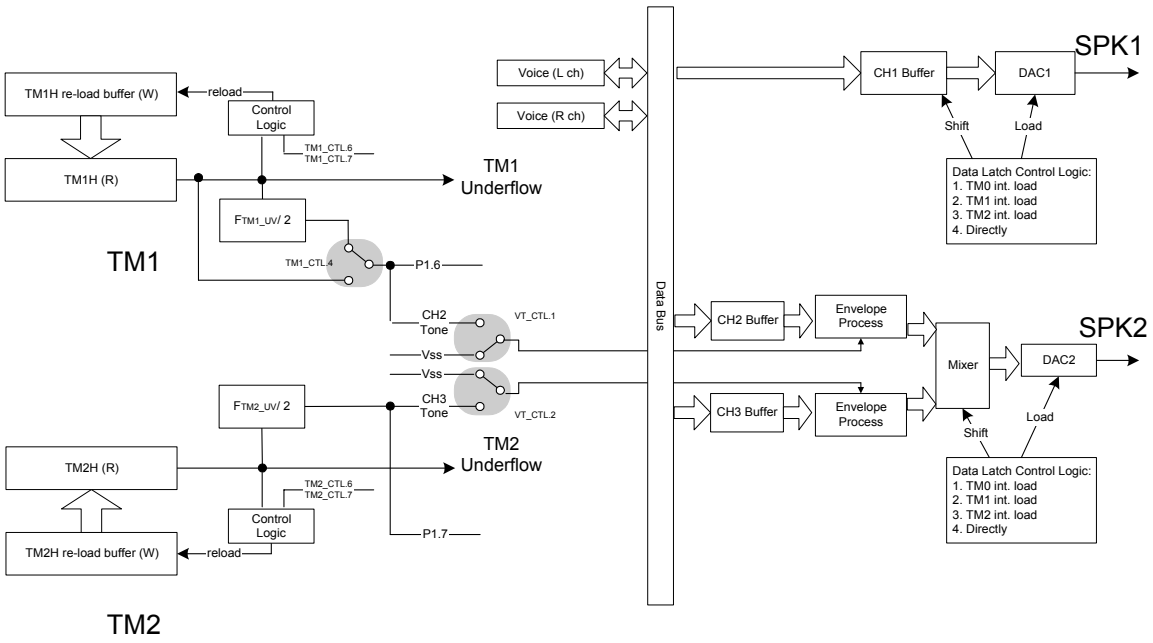
Temporary speech data output buffer. For playing a voice, the program could be coded as below:

```
LDA     PCM     ; Load PCM data into DAC buffer
STA     E1H     ; Latch 8-bit data (CH1) into DAC1 if directly mode is selected.
```

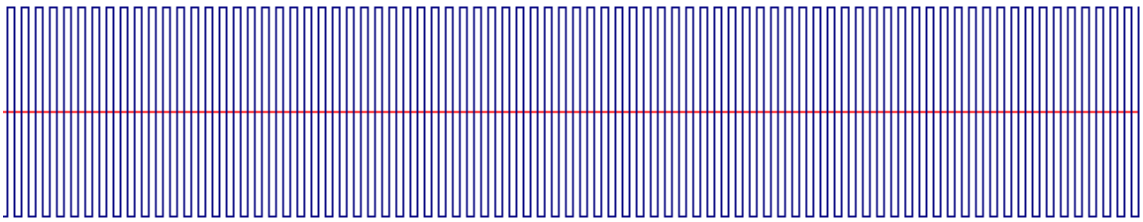
CH2, CH3 Buffer

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00E3H	CH2/ENV2	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	√	√
00E5H	CH3/ENV3	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	√	√

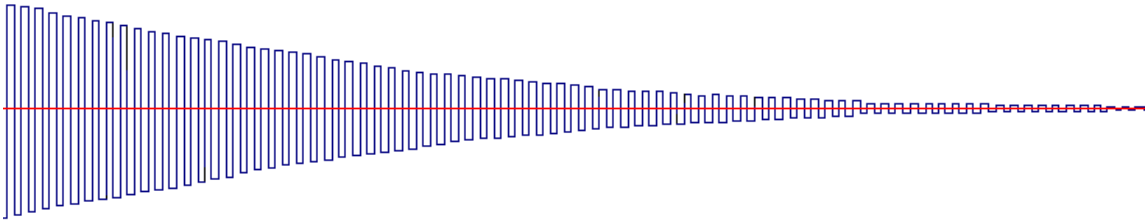
Temporary speech data output buffer.



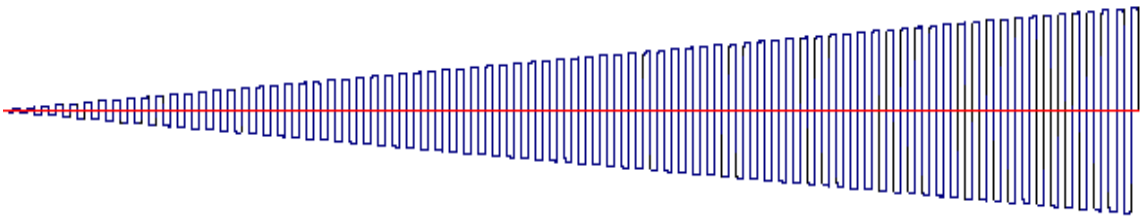
The CH2 and CH3 buffers could work as envelope setting registers that designed to control the output level of tone. When they are set to be 80H, the output is at the lowest – no any tone output. After stopping playing tone, the VT_CTL register should be set to voice mode and progress the fade out subroutine to avoid the noise burst. Changing the envelope of a tone can create various timbre of music. The waveform of normal square wave is like:



Program can create such waveforms as below through the envelope setting register.



(Envelope A)



(Envelope B)

The same tone with envelope A and B sounds very different. The tones with envelope-A sounds like piano and envelope-B sounds like harmonica.

DAC buffer transfer control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00F9H	DB_TC	-	TC6	TC5	TC4	-	TC2	TC1	TC0	√	√

TC6 ~ TC4 control Ch2/Ch3, TC2 ~ TC0 control Ch1. Before disable all the DAC output, user must progress the fade out subroutine to avoid the noise burst.

TC6: DAC2 enable control. 0: disable, 1:enable

TC5	TC4	DAC 2 buffer transfer control
0	0	Ch 2 / Ch3 buffer data transfer to DAC 2 after TM0 underflow
0	1	Ch 2 / Ch3 buffer data transfer to DAC 2 after TM1 underflow
1	0	Ch 2 / Ch3 buffer data transfer to DAC 2 after TM2 underflow
1	1	Ch 2 / Ch3 buffer data transfer to DAC 2 directly

TC2: DAC1 enable control. 0: disable, 1:enable

TC1	TC0	DAC 1 buffer transfer control
0	0	Ch 1 buffer data transfer to DAC 1 after TM0 underflow
0	1	Ch 1 buffer data transfer to DAC 1 after TM1 underflow
1	0	Ch 1 buffer data transfer to DAC 1 after TM2 underflow
1	1	Ch 1 buffer data transfer to DAC 1 directly

Voice/Tone control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00FAH	VT_CTL	-	-	-	-	-	VTS3	VTS2	-	√	√

Ch3, Ch2 voice/tone path control register.

VTS3: Ch 3 voice/tone control. 0: voice, 1:tone

VTS2: Ch 2 voice/tone control. 0: voice, 1:tone

DAC output current

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00FDH	DAC_DRV	-	-	-	-	-	-	DRV1	DRV0	√	√

Program can select the driving current of DAC output to fit different bipolar junction transistor for generating appropriate sound quality.

DRV1	DRV0	DAC output drive current (V_{DD} = 3.0V)
0	0	1.30 mA (default)
0	1	1.84 mA
1	0	2.60 mA
1	1	3.67 mA

Programming Notice

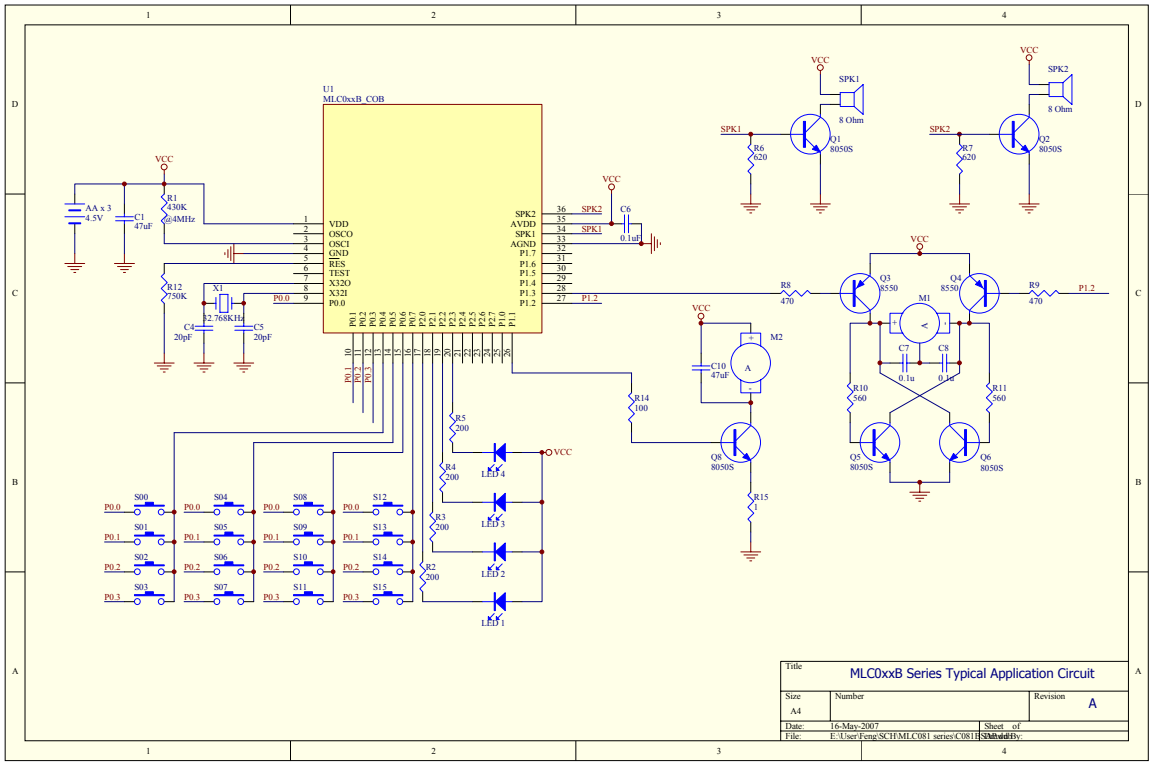
The status after different reset condition is listed below:

	Power on reset	CPU /RST pin reset
SRAM Data	Unknown	Unchanged
CPU Register	Unknown	Unknown
Special Function Register	Default value	Default value

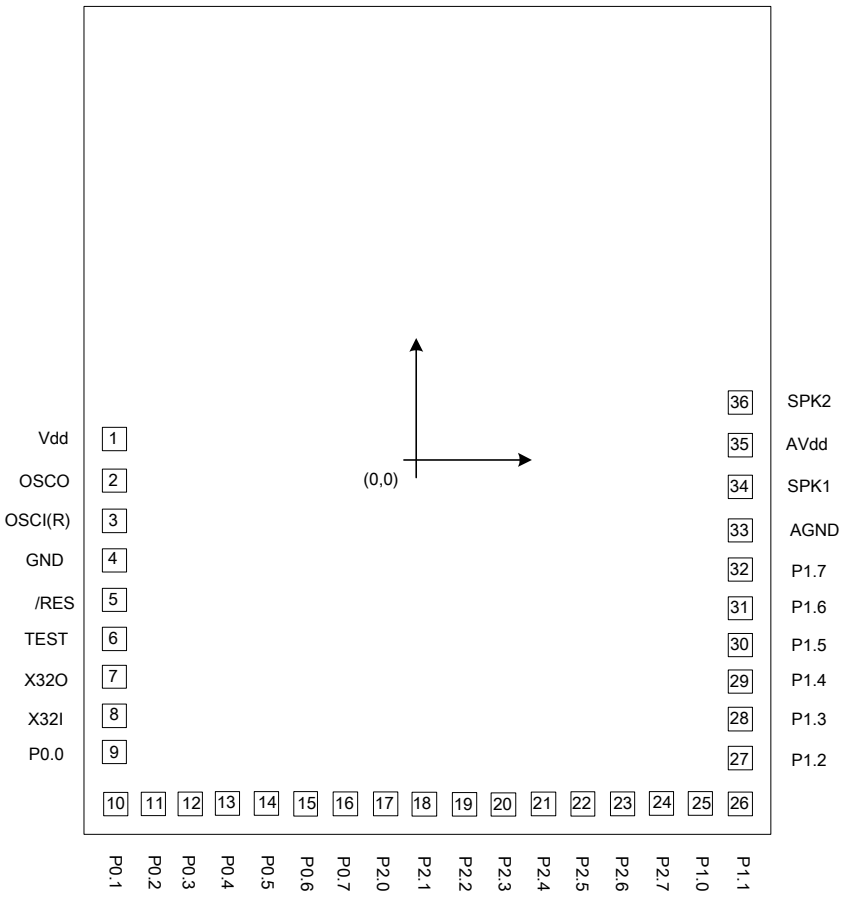
Mask Option

Clock source	Single / Dual
Fosc	RC / Crystal
WDT	Enable / Disable

Application Circuit



Pad Assignment



Absolute Maximum Rating

PARAMETER	RATING	UNIT
Supply Voltage to Ground Potential	-0.3 to +5.0	V
Applied Input / Output Voltage	-0.3 to +5.0	V
Power Dissipation	60	mW
Ambient Operating Temperature	0 to +70	°C
Storage Temperature	-55 to +150	°C

Note: Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

DC Characteristics

(V_{DD}-V_{SS} = 3.0 V, F_{OSC} = 4MHz, Ta = 25° C; unless otherwise specified)

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Op. Voltage	V_{DD}	-	2.4	-	5.5	V
Op. Current	I_{OP}	No load (Ext.-V) In normal operation	-	1.5	5.1	mA
Standby Current	I_{STB}	No load (Ext.-V)	-	1	3	μA
DAC output driving current	I_{DAC}	DAC_DRV = 00H	-	1.30	-	mA
		DAC_DRV = 01H	-	1.84	-	
		DAC_DRV = 02H	-	2.60	-	
		DAC_DRV = 03H	-	3.67	-	
Input High Voltage	V_{IH}	-	0.8 V_{DD}	-	V_{DD}	V
Input Low Voltage	V_{IL}	-	0	-	0.2V_{DD}	V
Port 0, 1, 2 drive current	I_{OH}	V_{OH} = 2.7V, V_{DD} = 3.0V	-	1.5	-	mA
Port 0, P1.4~1.7, P2.4~2.7 sink current	I_{OL0}	V_{OL} = 0.4V, V_{DD} = 3.0V	-	3.0	-	mA
Port 1.0~1.3, 2.0~2.3 sink current	I_{OL1}	V_{OL} = 0.4V, V_{DD} = 3.0V	-	9.0	-	mA
Internal Pull-high Resistor (L)	R_{PH0}	V_{IL} = 0V	-	350K	-	Ω
Internal Pull-high Resistor (S)	R_{PH1}	V_{IL} = 0V, port 0 only	-	50K	-	Ω

AC Characteristics

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
CPU Op. Frequency	F_{CPU}	RC/Crystal, V_{DD} = 3.0V	0.5	4	-	MHz
		RC/Crystal, V_{DD} = 5.0V	0.5	8	-	
Frequency Deviation by Voltage Drop for RC Oscillator	$\frac{\Delta f}{f}$	$\frac{f(3.0V) - f(2.4V)}{f(3.0V)}$	-	2	4	%
POR duration	T_{POR}	F_{OSC} = 4 MHz	10	15	50	mS

History:

V0.10: Original

V0.20: Add FFEDH on the memory map diagram (page 6)

V0.30: Correct the SRS function definition (page 17, set this bit to 1 should be shift right)

V0.40: Add the description about the output mode of I/O port (page 21)

V0.50: Modify the typing error of page 21 (change 100K to 350K)

V0.60: Modify some typing mistakes