矽源特科技
ChipSourceTek

NY8LP10A

8-bit 65C02 MCU with 6x30 LCD Driver, 16 I/O, & Buzzer Output

Version 1.4

Nov. 30, 2020

Revision History

Version	Date	Description	Modified Page
1.0	2019/11/21	Formal release.	-
1.1	2020/02/17	1. LVD function 2. DC characteristic Slow mode @3V = 15uA	4, 6 11
1.2	2020/06/01	1. Add registers description into whole chapters.	Page9~42
1.3	2020/08/04	1. Feature list updated. 2. Section 5.1 8MHz -> 4MHz. 3. Section 6.2 Timer2/Timer1/Timer0 -> Timer1/Timer0. 4. Add KS Interrupt. 5. Section 7.2 DPRAM Address and Size updated. 6. Section 8.2 LCD RAM table updated. 7. Section 12.2 FCPU = 32KHz -> 500KHz. 8. Add EL. 9. Add Key Strobe.	5, 7 10 16 18 25 27 49 41~42 37
1.4	2020/11/30	Add Note “It needs to be cautious to use OPMD to divide the clock frequency. Please refer to AP-Note 34 for details.” at Section 5.2.	12

Table of Contents

1. 概述

NY8LP10A 爲高性能 8 位元 65C02 微控制器附加 LCD 驅動和 Buzzer 播放功能，三組8 位元 timer / counter，16 根I/O。ROM 部份為崁入式 EPROM 架構的 OTP IC (One Time Programmable)。LCD 驅動單元除控制面板功能之外，還內建按鍵偵測功能。

MCU 為 CISC 架構易於編程和控制以及規劃到多種的應用。此外並提供多種工作模式 Slow mode, Standby mode 及 Halt mode (Sleep Mode) 來有效減少功耗。

2. 功能

- 寬廣的工作電壓範圍： 1.5V应用，1.1V~3.6V @ System clock ≤ 500KHz;
  3.0V应用，1.8V~3.6V @ System clock ≤ 4MHz。
- 16K-Byte OTP ROM。
- 128-Byte RAM。
- LCD 點數 (COM x SEG)：6 x 30。
- 16 GPIO，其中 7 根和 LCD SEG 共用。
- 雙時脈振盪：系統時鐘可自由選擇高速或低速。
  - 高速振盪: IOSC4M / IOSC2M / IOSC500K
  - 低速振盪: IOSC32K / XTAL32K。
- 內建高精準振盪線路(+/- 1.5%)。
- 四種工作模式可有效省電減少功耗：
  - Normal mode、Slow mode、Standby mode 及 Halt mode。
- Normal mode 下 CPU clock 速度可程式化：
  - 可設定為高速振盪的 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128。
- 3 組 8 位元 Timer，可應用於1通道或2通道Buzzer或 RFC 等等應用。
- 支援大多數 LCD 顯示：
  - 1/2, 1/3 bias。
  - 1/2, 1/3, 1/4, 1/5, 1/6 duty。
- 內建 Charge pump 升壓供應 LCD 顯示。
- 1.5V應用的內部RC振盪器模式，LCD幀速率最小值為42Hz。
- COM & SEG 可設定成 LED sink / drive 功能。
- LCD 支援 Matrix key 功能，input & output 起始位置可在組態中設定。
- Serial peripheral interface (SPI) Master mode 可存取 SPI 元件。
- EL驅動功能，可設定頻率和佔空比。

- RFC 功能，可用於温度、濕度偵測應用。
- 完整的系統保護，Watch-dog reset 看門狗重置功能及external reset pin 外部重置腳。
- 内建1.1V 或1.85V 低電壓偵測。
- 彈性化的 I/Os 設定：floating 輸入、pull-low 輸入、CMOS 輸出、open-drain 輸出。
- 紅外線載波頻率可供選擇，同時載波之極性也可以根據數據作選擇。
- 1 或 2 通道Buzzer。
- 7 種中斷模式。
- LCD 點數組合：

COMMON	SEGMENT	DOTS
6	30	180
5	31	155
4	32	128
3	32	96
2	32	64

1. GENERAL DESCRIPTION

NY8LP10A is a high-performance 8-bit 65C02 micro-controller with LCD driver and buzzer output, three sets of 8-bit timer/counter, 16 general I/Os. It's embedded EPROM architecture OTP (One Time Programmable). For LCD driver, it applies for the most common-used LCD panels and functions as a key strobe for further application.

The CISC MCU architecture is very easy to program and control, various applications can be easily implemented. Furthermore, in addition to the Slow mode, it offers the Standby mode and Halt mode (Sleep mode) to minimize power dissipation.

2. FEATURES

- Wide operating voltage range: ;

  For 1.5V application, 1.1V ~ 3.6V @ System clock ≤ 500KHz;

  For 3.0V application, 1.8V ~ 3.6V @ System clock ≤ 4MHz.
- 16KB OTP ROM.
- 128B RAM.
- LCD Dots (COM x SEG): 6 x 30.
- 16 GPIO, 7 shared from LCD SEG.
- Dual-clock oscillation: System clock can switch between high oscillation and low oscillation.
  - High OSC: IOSC4M / IOSC2M / IOSC500K.
  - Low OSC: IOSC32K / XTAL32K.
- Precisely embedded oscillator with build-in resistor (+/- 1.5%).
- Four kinds of operation mode to reduce system power consumption:
  - Normal mode, Slow mode, Standby mode and Halt mode.
- At Normal mode, CPU clock is software programmable.
  - 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128 of high oscillator (F_{FAOS}) frequency.
- Three 8-bit timers for 1-channel or 2-channel buzzer or other applications such as RFC.
- Support most of LCD panel types:
  - 1/2, 1/3 bias.
  - 1/2, 1/3, 1/4, 1/5, 1/6 duty.
- Charge pump for the LCD display power.
- For internal RC oscillator mode with 1.5v applications, the LCD frame rate minimum value is 42Hz
- LED sink/drive configuration supported through COM & SEG.
- Matrix key supported, and input & output starting pad can be selected by option
- Serial peripheral interface (SPI) Master mode for serial Flash/SRAM memory.

- EL driver block supported with various frequency and duty
- RFC-functioned block for the detection of humidity, temperature or other applications.
- Low voltage reset, watch-dog reset (by option) and external reset pin (by option) are all supported to protect the system.
- 1.1V or 1.85V LVD flag for low battery detection.
- Flexible I/Os maximum with optional function: floating input, pull-low input, CMOS output, open-drain output.
- Infrared output: optional IR carrier frequency and optional data high/low IR output supported.
- 1-channel or 2-channel buzzer
- 7 interrupt modes supported.
- Possible LCD COM and SEG combination:

COMMON	SEGMENT	DOTS
6	30	180
5	31	155
4	32	128
3	32	96
2	32	64

3. BLOCK DIAGRAM

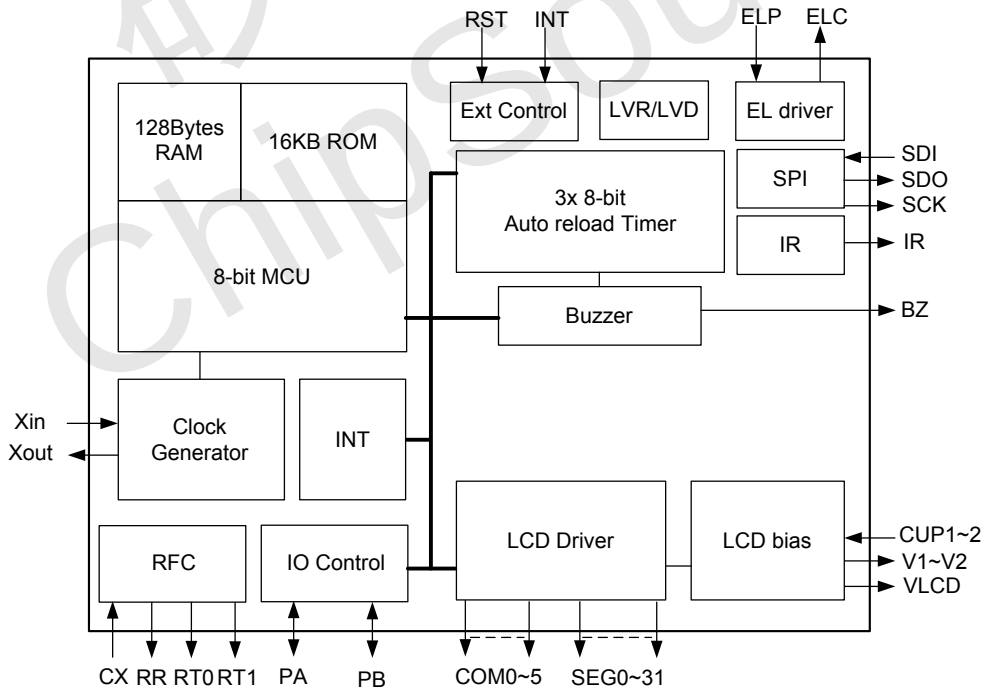

Fig.3-1: The block diagram

4. PAD DESCRIPTION

Pad Name	ATTR.	Description
VDD	P	Positive supply power.
VSS	P	Negative supply power.
PA0/Xin	I/O	Bit 0 for Port A, or input of XTAL32K.
PA1/Xout	I/O	Bit 1 for Port A, or output of XTAL32K.
PA2/INT/Vpp	I/O	Bit 2 for Port A, or external interrupt input, or Vpp for programming.
PA3/RST/Mode	I/O	Bit 3 for Port A, or external reset input, or select programming mode.
PA4/CX	I/O	Bit 4 for Port A, or input of RFC function.
PA5/RR	I/O	Bit 5 for Port A, or output of RFC function.
PA6/RT0	I/O	Bit 6 for Port A, or output of RFC function.
PA7/RT1	I/O	Bit 7 for Port A, or output of RFC function.
PB0/BZ	I/O	Bit 0 for Port B, or buzzer output.
SEG0/PB1	I/O	LCD segment 0, Bit 1 for Port B. (can be used as key strobe input)
SEG1/PB2/ELP/SDA	I/O	LCD segment 1, Bit 2 for Port B, or charging signal of EL driver, or serial data input at programming mode. (can be used as key strobe input)
SEG2/PB3/ELC/SCL	I/O	LCD segment 2, Bit 3 for Port B, or discharging signal of EL driver, or serial clock input at programming mode (can be used as key strobe input)
SEG3/PB4/IR	I/O	LCD segment 3, Bit 4 for Port B, or IR output.
SEG4/PB5/SCK	I/O	LCD segment 4, Bit 5 for Port B, or clock output of SPI.
SEG5/PB6/SDI	I/O	LCD segment 5, Bit 6 for Port B, or data input (MISO) of SPI.
SEG6/PB7/SDO	I/O	LCD segment 6, Bit 7 for Port B, or data output (MOSI) of SPI.
SEG7~29	O	LCD segment 7~29.
COM0~5	O	LCD common 0~5 (COM4~5 can be used as SEG31~30).
V1~2, VLCD	P	LCD supply power.
CUP1~2	I/O	Auxiliary capacitor pins for voltage pumping.

Total：52 Pins

Legend: I = Input, O = Output, P = Power, A = Analog

5. Operation Modes

5.1 Clock Source

Because NY8LP10A is a dual-clock IC, there are fast oscillator (F_{FAOS}) and slow oscillator (F_{SLOW}) that can be selected as system oscillation (F_{CPU}). The fast oscillator which could be used as F_{FAOS} is internal high RC oscillator: IOSC4M. The slow oscillators which could be used as F_{SLOW} are internal low RC oscillator (IOSC32K) or external low crystal oscillator (XTAL32K). Users can choose the clock sources by programming its option based on the application.

To utilize the precise timing application, two pins (Xin & Xout) are needed to connect with external crystal module and set the corresponding option for 32KHz crystal. To match the high-speed application, it provides up to 4MHz for F_{CPU} and no additional pins are needed.

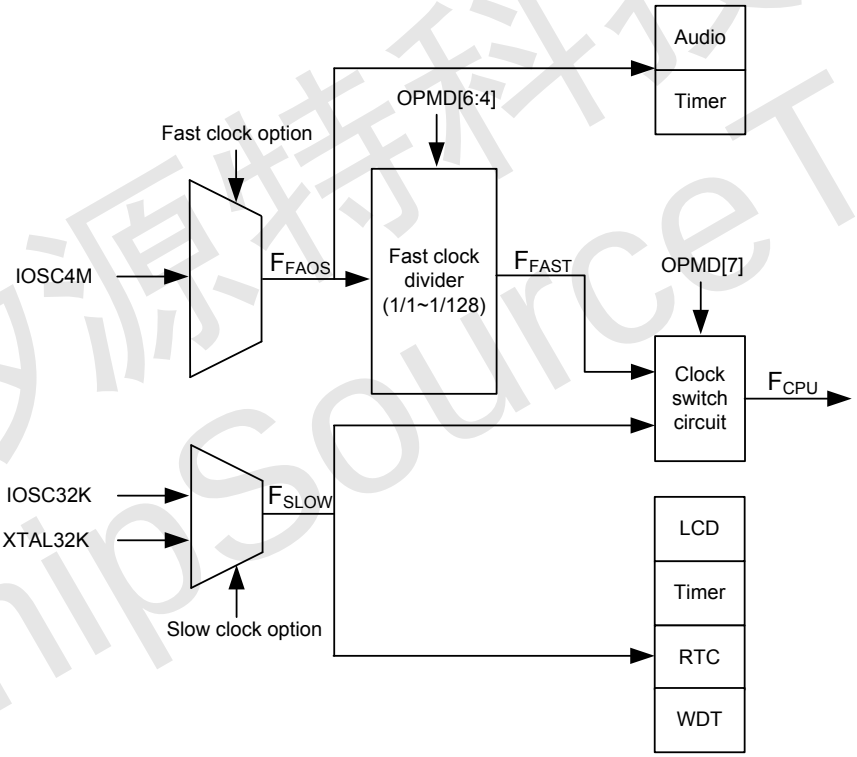

Figure 5-1 Oscillation Configuration

The fast frequency (F_{FAST}) is divided from F_{FAOS} through control register OPMD[6:4]: 1/1~1/128. The F_{SLOW} is the slow frequency and the source of Real-Time-Clock (RTC). The RTC is the clock source of 14-bit divider, ranges from 16KHz to 2Hz, and it generates multiple clocks to apply to the LCD module, LCD power charge pump block, watch dog timer, or EL block, etc. Therefore user is suggested to enable the RTC (OPMD[3]) when CPU is busy. Besides, the RTC counter can be cleared by writing 0 to OPMD[2], and which is always read as high.

Mode	Normal mode	Slow mode	Standby mode	Halt mode
F_{CPU}	ON (F_{FAST})	ON (F_{SLOW})	OFF	OFF
F_{FAOS}	ON	OFF	OFF	OFF
F_{FAST}	$F_{FAOS}/2^N$	OFF	OFF	OFF
F_{SLOW}	ON/OFF	ON	ON	OFF
RTC	ON/OFF	ON	ON	OFF
LCD	ON/OFF	ON/OFF	ON/OFF	OFF
Wake-up Source	--	--	- Key change - Timer2/Timer1/ Timer0 Interrupt (based on F_{SLOW}) - FT/ST Interrupt - External Interrupt	- Key change - External Interrupt

NY8LP10A provide four kinds of operating mode to tailor all kinds of application and save power consumptions. These operating modes are Normal mode, Slow mode, Standby mode and Halt mode. Normal mode is designated for high-speed operating mode. Slow mode is designated for low-speed mode in order to save power consumption. At Standby mode, NY8LP10A will stop almost all operations except Timer2 /Timer1 /Timer0 /FT /ST (based on F_{SLOW}) in order to wake up periodically. At Halt mode, NY8LP10A will sleep until key change or external interrupt occurs. User can set the control register OPMD to swap Normal/Slow mode and the control register SLP to enter Standby/Halt mode. The block diagram of four operating modes is described in Figure 2-2.

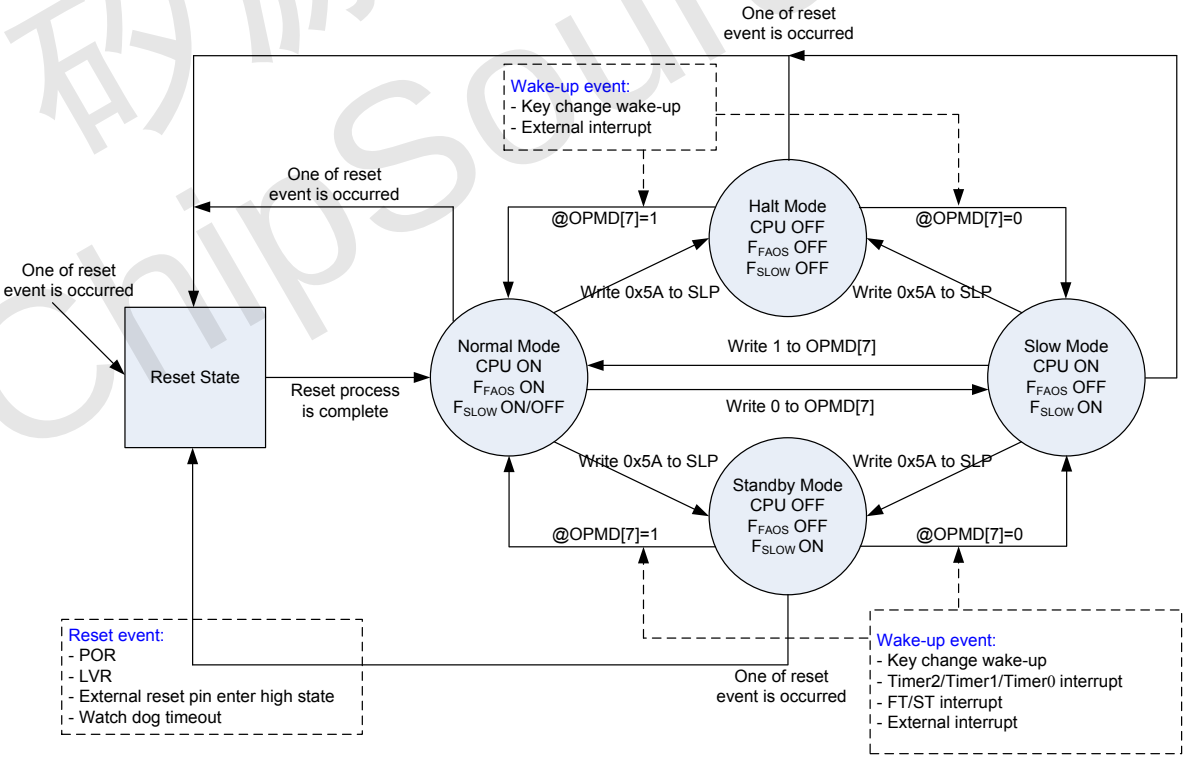

Figure 5-2: Four Operating Modes

5.2 Normal Mode

After any reset event is occurred and reset process is complete, NY8LP10A will enter Normal mode. At Normal mode, F_{FAOS} is selected as system oscillation in order to provide highest performance and its power consumption will be the largest among four operating modes.

- Instruction execution is based on F_{FAST} and all peripheral modules may be active according to corresponding module enable bit.
- F_{FAST} is divided from the F_{FAOS} (1/1 ~ 1/128).
- F_{SLOW} is enabled, or disabled according to application.
- IC can switch to Slow mode by writing 0 to the bit7 of control register OPMD ($17[7]).
- IC can switch to Standby mode or Halt mode by programming control register SLP ($15) with 0x5A.

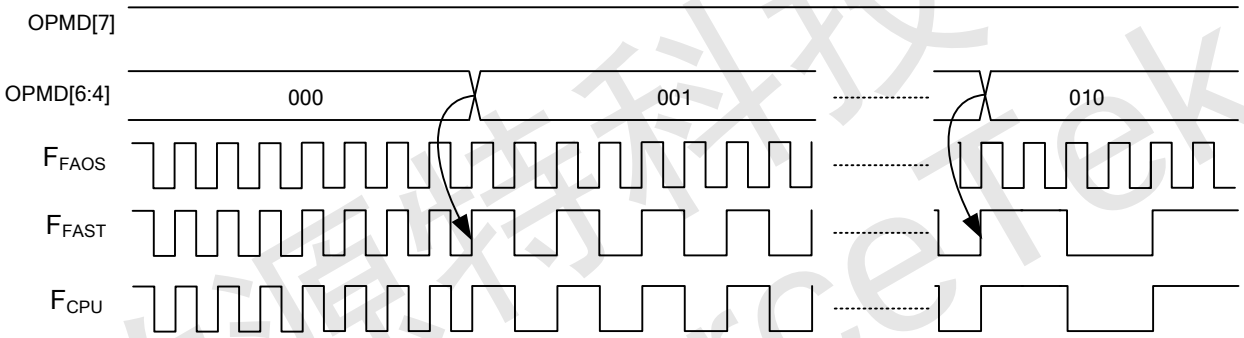

Figure 5-3: The procedure of dividing CPU clock

Users have to set the control register OPMD to the relative setting as the following table.

Addr.	Name	R/W	Bit	Data	Description	Default
$17	OPMD	W	[2]	0	Write 0 to clear RTC counter (read as high)	x
		R/W	[3]	1/0	RTC Enable/Disable	Enable
			[6:4]	000	$F_{FAST} = F_{FAOS}/1$	F_{FAOS} /1
				001	$F_{FAST} = F_{FAOS}/2$	
				010	$F_{FAST} = F_{FAOS}/4$	
				011	$F_{FAST} = F_{FAOS}/8$	
				100	$F_{FAST} = F_{FAOS}/16$	
				101	$F_{FAST} = F_{FAOS}/32$	
				110	$F_{FAST} = F_{FAOS}/64$	
				111	$F_{FAST} = F_{FAOS}/128$	
			[7]	1/0	$F_{CPU} = F_{FAST}/F_{SLOW}$ (32KHz)	F_{FAST}

Note1: It needs to be cautious to use OPMD to divide the clock frequency. Please refer to AP-Note 34 for details.

5.3 Slow Mode

NY8LP10A will enter Slow mode by writing 0 to the bit7 of control register OPMD ($17[7]). At Slow mode, F_{SLOW} is selected as system oscillation in order to save power consumption but still keep IC running. However, the F_{FAOS} should be turned off permanently if the mode won't be swapped to Normal mode. When switching to Normal mode, the fast clock source must wait about 512 cycles (~128us@4MHz) for being stable. It is strongly recommended that IC should switch to Slow mode after F_{SLOW} being stable (~120us@IOSC32K or ~30ms@XTAL32K). Once setting control register SLP ($15) with 0x5A, it will turn to Standby mode or Halt mode, and the FCPU will be stopped until the wake-up signal occurs.

- Instruction execution is based on FSLOW and all peripheral modules may be active according to corresponding module enable bit.
- FFAOS can be turned off by writing 0 to OPMD[7].
- IC can switch to Standby mode or Halt mode by programming control register SLP ($15) with 0x5A.
- IC can switch to Normal mode by writing 1 to OPMD[7].

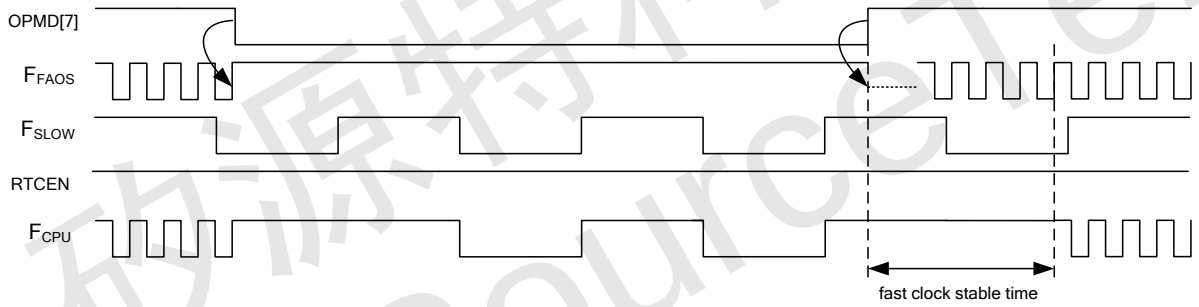

Figure 5-4: The procedure of switching operation modes

5.4 Standby Mode

By setting control register SLP ($15) with 0x5A, the operation mode will turn to Standby mode if RTC is enabled (OPMD[3] = 1), while the previous mode is either Normal mode or Slow mode. At Standby mode, the F_{FAOS} is shut down and the F_{SLOW} is kept to supply the clock for LCD display, etc.

NY8LP10A supports two wake-up methods to leave out of Standby mode, the difference between I/O pads and its data registers (key change), the occurrence of each interrupt, So before entering Standby mode, users have to keep in mind to store the current input port statuses into port registers,. If the system is waked up, the succeeding instructions after writing SLP register will be executed after the clock source stable time. The stable time of fast clock source should wait about 512 cycles (~128us@4MHz), and the stable time of IOSC32K is about 4 cycles (~120us@32KHz), the stable time of XTAL32K is about 1024 cycles (~30ms@32KHz).

If the IC is waked up from Standby mode by a reset pin, it goes into reset procedure.

- Instruction execution is stop and some peripheral modules may be active according to corresponding module enable bit.
- FFAOS can be shut down by writing 0x5A to register SLP ($15).
- The FSLOW is still active and running.
- IC can being waked up from Standby mode if any of (a) key change wake-up (refer to 7.1 I/O Ports), (b) Timer2/Timer1/Timer0 (based on FSLOW) interrupt, (c) FT/ST interrupt, (d) external interrupt.
- After being waked up from Standby mode, IC will return to Normal mode if OPMD[7] = 1, or Slow mode if OPMD[7] = 0.

The relative control registers are shown as the following tables

Addr.	Name	R/W	Bit	Data	Description	Default
$15	SLP	W	[7:0]		Write 0x5A to sleep	xx

5.5 Halt mode

By setting control register SLP ($15) with 0x5A, the operation mode will turn to Halt mode if RTC is disabled (OPMD[3] = 0), while the previous mode is either Normal mode or Slow mode. Halt mode is also known as Sleep mode. As implied by the name, the IC falls asleep and the system clock is completely turned off, so all the IC functions are halted and it minimizes the power consumption.

At Halt mode, both of the F_{FAOS} and the F_{SLOW} are shut down and waked up by key change or external interrupt. So before entering Halt mode, users have to keep in mind to store the current input port statuses into port registers. For avoiding awaking Halt mode wrongly,-, and the data register (PX) must be cleared to low. If the system is waked up, the succeeding instructions after writing SLP register will be executed after the clock source stable time.

If the IC is waked up from the standby mode by a reset pin, it goes into the reset procedure.

- Instruction execution is stop and all peripheral modules are disabled.
- FFAOS and FSLOW are both disabled automatically.
- IC can being waked up from Halt mode if any of (a) Key change wake-up (refer to 7.1 I/O Ports) or (b) external interrupt is happened.
- After being waked up from Halt mode, IC will return to Normal mode if OPMD[7] = 1, or Slow mode if OPMD[7] = 0.

The relative control registers are shown as the following tables:

Addr.	Name	R/W	Bit	Data	Description	Default
$15	SLP	W	[7:0]		Write 0x5A to sleep	xx

6. System Control

6.1 Reset System

For the NY8LP10A IC, the reset procedure needs at least 125ms to deal with initialization process. In addition, 4 conditions will cause the reset procedure to be triggered, described in next sections. The reset initialization procedure is shown in Figure 3-1.

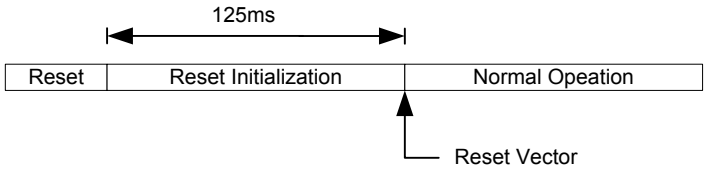

Figure 6-1: The reset initialization procedure

6.1.1 Power-On Reset (POR)

After power-on, the power-on reset initialization will automatically be set out. After the system leaves the reset initialization procedure, it enters the normal operation and the program counter (PC) starts at the reset vector.

6.1.2 Low Voltage Reset (LVR)

When the system enters the normal operation, the power supply voltage must be kept in an effective working voltage range. When the power supply voltage is lower than the effective operating voltage range, the system can't work properly. When the detector detects a harmful low voltage supply, it will cause a low voltage reset.

6.1.3 External Reset Pin (by option)

The external reset pin is always pulled-low with strong or weak resistor controlled by option. Generally, when the reset pin rises to high, it generates an external reset.

6.1.4 Watch-Dog Timer Reset (WDTR) (by option)

To recover from program malfunction, the NY8LP10A IC supports an embedded watch-dog timer reset by option. The WDTR function is based on Real-Time-Clock, and always works with the program executing. Users have to clear the WDT ($16) periodically to prevent from timing up with a reset generation. Typically, the minimum time-up period of the WDT is about 1.5s.

Users can write 0xA5 to the control register WDGC to clear WDT, The relative control registers are shown as the following tables.

Addr.	Name	R/W	Bit	Data	Description	Default
$16	WDGC	W	[7:0]		Write 0xA5 to clear watchdog timer	xx

6.1.5 Low Voltage Detector (LVD)

To monitor the voltage supply, the NY8LP10A also provides low voltage detector (LVD) function. The LVD is fixed as 1.1V for 1.5V application and 1.85V for 3.0V application. If LVD is enabled, user can read back LVD status, which will go high when the power supply is lower than LVD level. The setting of LVD is shown as below.

Addr.	Name	R/W	Bit	Data	Description	Default
$1C	LVD	R/W	[4]	1/0	LVD Enable/Disable	Disable
		R	[7]	1/0	LVD status: VDD<LVD level/VDD>LVD level	x

6.2 Interrupts

The interrupt event can be a fixed interval of the Timer2/Timer1/Timer0, the fast real timer (FT), the slow real timer (ST),a random period triggered by the external interrupt pin (INT) or the occurrence of key strobe(KSB) .The Timer1/Timer0 can also be selected as one of the sample rate for audio playing, and the KSBF arises as a key strobe pulse occurs. There are two real timers (FT & ST) in the NY8LP10A IC, which function as long as it isn't in the halt mode. NY8LP10A provide 8 fixed intervals from the real timer for FT, ranged from 16Hz to 16KHz, and it provide 8 fixed intervals from the real timer for ST, ranged from 256Hz to 2Hz. The interrupt events have to be cleared by users after entering the interrupt routine.

While any of hardware interrupts is occurred, the corresponding bit of interrupt flag register INTF will be set to 1. This bit will not be clear until users write 0 to this bit. Therefore user can obtain information of which event causes hardware interrupt by polling register INTF even if interrupt enable flag register IEF is Disable. The detailed settings of interrupt mode and flag are shown as the following tables.

Addr.	Name	R/W	Bit	Data	Description	Default
$0C	IEF	R/W	[0]	1/0	Timer2 Interrupt Enable/Disable	Disable
			[1]	1/0	Timer1 Interrupt Enable/Disable	Disable
			[2]	1/0	Timer0 Interrupt Enable/Disable (or Timer0 stop@Timer0 stop enable)	Disable
			[3]	1/0	FT Interrupt Enable/Disable	Disable
			[4]	1/0	ST Interrupt Enable/Disable	Disable
			[5]	1/0	EXT Interrupt Enable/Disable	Disable
			[7]	1/0	KEY_STROBE INT Enable/Disable	Disable
$0D	INTF	R	[0]	1/0	Read Timer2 Interrupt Flag	0
			[1]	1/0	Read Timer1 Interrupt Flag	0
			[2]	1/0	Read Timer0 Interrupt Flag (or Timer0 stop@Timer0 stop enable)	0
			[3]	1/0	Read FT Interrupt Flag	0
			[4]	1/0	Read ST Interrupt Flag	0
			[5]	1/0	Read EXT Interrupt Flag	0
			[7]	1/0	Read KEY_STROBE INT Flag	0
		W	[0]	0	Clear Timer2 Interrupt Flag	0
			[1]	0	Clear Timer1 Interrupt Flag	0

Addr.	Name	R/W	Bit	Data	Description	Default
			[2]	0	Clear Timer0 Interrupt Flag (or Timer0 stop@Timer0 stop enable)	0
			[3]	0	Clear FT Interrupt Flag	0
			[4]	0	Clear ST Interrupt Flag	0
			[5]	0	Clear EXT Interrupt Flag	0
			[7]	0	Clear KEY_STROBE INT Flag	0
$0E	NMI	R/W	[0]	1/0	Timer2 Interrupt Flag be NMI/IRQ	IRQ
			[1]	1/0	Timer1 Interrupt Flag be NMI/IRQ	IRQ
			[2]	1/0	Timer0 Interrupt Flag be NMI/IRQ (or Timer0 stop@Timer0 stop enable)	IRQ
			[3]	1/0	FT Interrupt Flag be NMI/IRQ	IRQ
			[4]	1/0	ST Interrupt Flag be NMI/IRQ	IRQ
			[5]	1/0	EXT Interrupt Flag be NMI/IRQ	IRQ
			[7]	1/0	KEY_STROBE INT Flag be NMI/IRQ	IRQ
$0F	RTC	R/W	[2:0]	000	FT Interrupt = RT[10] (16Hz, F_{SLOW}/2048)	RT[4]
				001	FT Interrupt = RT[8] (64Hz, F_{SLOW}/512)	
				010	FT Interrupt = RT[6] (256Hz, F_{SLOW}/128)	
				011	FT Interrupt = RT[4] (1KHz, F_{SLOW}/32)	
				100	FT Interrupt = RT[3] (2KHz, F_{SLOW}/16)	
				101	FT Interrupt = RT[2] (4KHz, F_{SLOW}/8)	
				110	FT Interrupt = RT[1] (8KHz, F_{SLOW}/4)	
				111	FT Interrupt = RT[0] (16KHz, F_{SLOW}/2)	
			[5:3]	000	ST Interrupt = RT[13] (2Hz, F_{SLOW}/16384)	RT[13]
				001	ST Interrupt = RT[12] (4Hz, F_{SLOW}/8192)	
				010	ST Interrupt = RT[11] (8Hz, F_{SLOW}/4096)	
				011	ST Interrupt = RT[10] (16Hz, F_{SLOW}/2048)	
				100	ST Interrupt = RT[9] (32Hz, F_{SLOW}/1024)	
				101	ST Interrupt = RT[8] (64Hz, F_{SLOW}/512)	
				110	ST Interrupt = RT[7] (128Hz, F_{SLOW}/256)	
				111	ST Interrupt = RT[6] (256Hz, F_{SLOW}/128)	
			[6]	1/0	EXT Interrupt takes place at Rising/Falling edge	Rising
$14	KSB	R/W	[5]	0	KEY_STROBE INT = Key strobe occurs	0
				1	KEY_STROBE INT = Each key strobe scanning cycle	

Note: It is strongly recommended to set Timer2, Timer1, Timer0, FT, ST, KSB, external interrupt, control register before enabling interrupt, otherwise interrupt may be falsely triggered.

For example, if a Timer2 interrupt occurs, the IC pushes the program counter (PC) to stack (STK), and jump to the interrupt vector ($7E0) automatically. In the interrupt sub-routine, user must store the accumulator (ACC), register X/Y (X/Y), and draw the Y, X, and ACC back before Timer2 sub-routine being finished. With the return instruction executes, the interrupt routine is finished. The IC pops STK back to the PC, and back to the original track of the program. The addresses for each interrupt mode are described in below table.

Addr.	Interrupt Vector	IRQ Priority
$7E0	Timer2 Interrupt	1 (highest)
$7E2	Timer1 Interrupt	2
$7E4	Timer0 Interrupt	3
$7E6	FT Interrupt	4
$7E8	ST Interrupt	5
$7EA	EXT Interrupt	6
$7EC		7
$7EE	KS Interrupt	8 (lowest)

The Timer2 interrupt sub-routine is shown below.

```
V-IRQ-Timer2:
PHA
PHX
PHY
……
PLY
PLX
PLA

RTI
```

The NY8LP10A IC supports two types of interrupt mode, IRQ and NMI (Non-Maskable Interrupt). The IRQ mode is sensed by **level-trigger** event, which means the continued interrupt event will be held until the current is finished. Even if two interrupt events come up simultaneously, the priority of each interrupt decides that the event with higher priority defined by IC itself will be enabled. The NMI mode is sensed by **edge-trigger** event, if an event is coming up with the others at the same time. There is the only one with highest priority defined by software will be enabled and the others may be falsely omitted, **so it is strongly recommended to enable ONLY ONE of NMI ($0E)**.

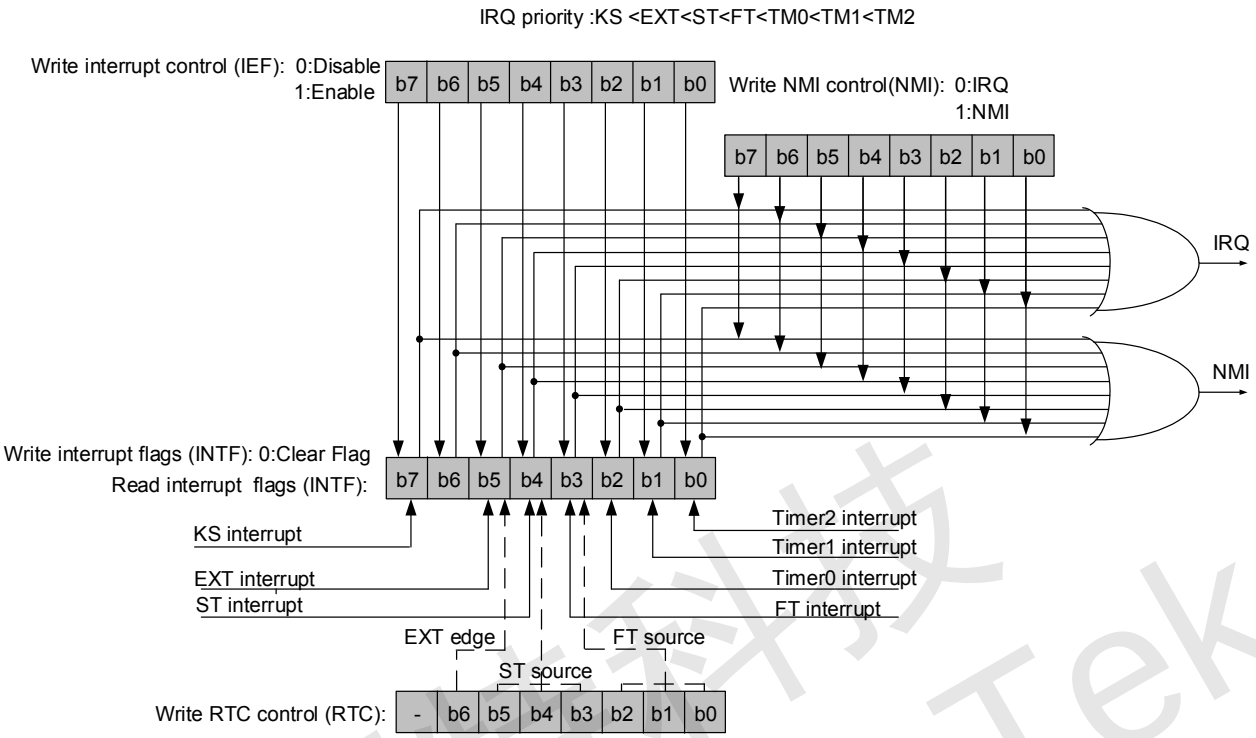

Figure 6-2: The structure of IRQ & NMI

7. Address Mapping

In the NY8LP10A IC, SFR contains 64 bytes, RAM contains 128 bytes for zero page register (ZP) and stack (STK). Moreover, there are 16K bytes ROM for program data. The following three sections describe the detail about special function register (SFR), RAM and ROM configuration of the NY8LP10A.

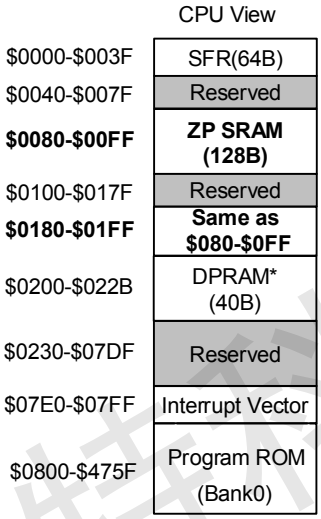

Figure 7-1: The structure of address mapping for NY8LP10A

7.1 Control Register Description

The special function register (SFR) is assigned to use the dedicated address ranged $0000 to $0039. Users can program these registers to fit their applications. The SFR table is shown below.

Addr.	Name	R/W	Bit	Data	Description	Default
$00	TM0D	R	[7:0]		Read the Timer0 counting data[7:0]	xx
		W	[7:0]		Preload Timer0 data[7:0]	xx
$01	TM0C	R/W	[3:0]	000x	Timer0 clock = CX	BT[0]
				001x	Timer0 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer0 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer0 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer0 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer0 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer0 clock = BT[6] (F_{FAOS}/128)	
				1001	Timer0 clock = BT[5] (F_{FAOS}/64)	
				1010	Timer0 clock = BT[4] (F_{FAOS}/32)	
				1011	Timer0 clock = BT[3] (F_{FAOS}/16)	
				1100	Timer0 clock = BT[2] (F_{FAOS}/8)	
				1101	Timer0 clock = BT[1] (F_{FAOS}/4)	
				1110	Timer0 clock = BT[0] (F_{FAOS}/2)	
				1111	Timer0 clock = F_{FAOS}	
			[4]	1/0	Timer0 Reload/One shot	1

Addr.	Name	R/W	Bit	Data	Description	Default
			[5]	1/0	Tone0 Enable/Disable	Enable
			[7:6]	00	Timer0 clock stop mode OFF	OFF
				01	Timer0 clock stopped by Timer2 overflow	
				10	Timer0 clock stopped by a full cycle of CX	
				11	Timer0 clock stopped by a full cycle of Timer2 clock	
$02	TM0EN	R/W	[0]	1/0	Timer0 Enable/Disable	Disable
$04	TM1D	R	[7:0]		Read the Timer1 counting data[7:0]	xx
		W	[7:0]		Preload Timer1 data[7:0]	xx
$05	TM1C	R/W	[3:0]	000x	Timer1 clock = TM0D[7]	BT[0]
				001x	Timer1 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer1 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer1 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer1 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer1 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer1 clock = BT[6] (F_{FAOS}/128)	
				1001	Timer1 clock = BT[5] (F_{FAOS}/64)	
				1010	Timer1 clock = BT[4] (F_{FAOS}/32)	
				1011	Timer1 clock = BT[3] (F_{FAOS}/16)	
				1100	Timer1 clock = BT[2] (F_{FAOS}/8)	
				1101	Timer1 clock = BT[1] (F_{FAOS}/4)	
				1110	Timer1 clock = BT[0] (F_{FAOS}/2)	
				1111	Timer1 clock = F_{FAOS}	
			[4]	1/0	Timer1 Reload/One shot	1
$06	TM1EN	R/W	[0]	1/0	Timer1 Enable/Disable	Disable
$08	TM2D	R	[7:0]		Read the Timer2 counting data[7:0]	xx
		W	[7:0]		Preload Timer2 data[7:0]	xx
$09	TM2C	R/W	[3:0]	000x	Timer2 clock = TM1D[7]	BT[2]
				001x	Timer2 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer2 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer2 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer2 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer2 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer2 clock = BT[8] (F_{FAOS}/512)	
				1001	Timer2 clock = BT[7] (F_{FAOS}/256)	
				1010	Timer2 clock = BT[6] (F_{FAOS}/128)	
				1011	Timer2 clock = BT[5] (F_{FAOS}/64)	
				1100	Timer2 clock = BT[4] (F_{FAOS}/32)	
				1101	Timer2 clock = BT[3] (F_{FAOS}/16)	
				1110	Timer2 clock = BT[2] (F_{FAOS}/8)	
				1111	Timer2 clock = BT[1] (F_{FAOS}/4)	
			[4]	1/0	Timer2 Reload/One shot	1
$0A	TM2EN	R/W	[0]	1/0	Timer2 Enable/Disable	Disable
$0C	IEF	R/W	[0]	1/0	Timer2 Interrupt Enable/Disable	Disable
			[1]	1/0	Timer1 Interrupt Enable/Disable	Disable
			[2]	1/0	Timer0 Interrupt Enable/Disable (or Timer0 stop@Timer0 stop enable)	Disable

Addr.	Name	R/W	Bit	Data	Description	Default
			[3]	1/0	FT Interrupt Enable/Disable	Disable
			[4]	1/0	ST Interrupt Enable/Disable	Disable
			[5]	1/0	EXT Interrupt Enable/Disable	Disable
			[7]	1/0	KEY_STROBE Interrupt Enable/Disable	Disable
$0D	INTF	R	[0]	1/0	Read Timer2 Interrupt Flag	0
			[1]	1/0	Read Timer1 Interrupt Flag	0
			[2]	1/0	Read Timer0 Interrupt Flag (or Timer0 stop@Timer0 stop enable)	0
			[3]	1/0	Read FT Interrupt Flag	0
			[4]	1/0	Read ST Interrupt Flag	0
			[5]	1/0	Read EXT Interrupt Flag	0
			[7]	1/0	Read KEY_STROBE INT Flag	0
		W	[0]	0	Clear Timer2 Interrupt Flag	0
			[1]	0	Clear Timer1 Interrupt Flag	0
			[2]	0	Clear Timer0 Interrupt Flag (or Timer0 stop@Timer0 stop enable)	0
			[3]	0	Clear FT Interrupt Flag	0
			[4]	0	Clear ST Interrupt Flag	0
			[5]	0	Clear EXT Interrupt Flag	0
			[7]	0	Clear KEY_STROBE INT Flag	0
$0E	NMI	R/W	[0]	1/0	Timer2 Interrupt Flag be NMI/IRQ	IRQ
			[1]	1/0	Timer1 Interrupt Flag be NMI/IRQ	IRQ
			[2]	1/0	Timer0 Interrupt Flag be NMI/IRQ (or Timer0 stop@Timer0 stop enable)	IRQ
			[3]	1/0	FT Interrupt Flag be NMI/IRQ	IRQ
			[4]	1/0	ST Interrupt Flag be NMI/IRQ	IRQ
			[5]	1/0	EXT Interrupt Flag be NMI/IRQ	IRQ
			[7]	1/0	KEY_STROBE INT Flag be NMI/IRQ	IRQ
$0F	RTC	R/W	[2:0]	000	FT Interrupt = RT[10] (16Hz, F_{SLOW}/2048)	RT[4]
				001	FT Interrupt = RT[8] (64Hz, F_{SLOW}/512)	
				010	FT Interrupt = RT[6] (256Hz, F_{SLOW}/128)	
				011	FT Interrupt = RT[4] (1KHz, F_{SLOW}/32)	
				100	FT Interrupt = RT[3] (2KHz, F_{SLOW}/16)	
				101	FT Interrupt = RT[2] (4KHz, F_{SLOW}/8)	
				110	FT Interrupt = RT[1] (8KHz, F_{SLOW}/4)	
				111	FT Interrupt = RT[0] (16KHz, F_{SLOW}/2)	
			[5:3]	000	ST Interrupt = RT[13] (2Hz, F_{SLOW}/16384)	RT[13]
				001	ST Interrupt = RT[12] (4Hz, F_{SLOW}/8192)	
				010	ST Interrupt = RT[11] (8Hz, F_{SLOW}/4096)	
				011	ST Interrupt = RT[10] (16Hz, F_{SLOW}/2048)	
				100	ST Interrupt = RT[9] (32Hz, F_{SLOW}/1024)	
				101	ST Interrupt = RT[8] (64Hz, F_{SLOW}/512)	
				110	ST Interrupt = RT[7] (128Hz, F_{SLOW}/256)	
				111	ST Interrupt = RT[6] (256Hz, F_{SLOW}/128)	
			[6]	1/0	EXT Interrupt takes place at Rising/Falling edge	Rising
$14	KSB	R/W	[3:0]		Key Strobe output SEG select	0000
			[4]	1/0	All/One SEGs selected as key strobe output	ONE

Addr.	Name	R/W	Bit	Data	Description	Default
			[5]	0	KEY_STROBE INT = Key strobe occurs	0
				1	KEY_STROBE INT = Each key strobe scanning cycle	
			[7:6]	00	KEY_STROBE Scan Rate = 256Hz (FSLOW /128)	256Hz
				01	KEY_STROBE Scan Rate = 512Hz (FSLOW /64)	
				10	KEY_STROBE Scan Rate = 1KHz (FSLOW /32)	
				11	KEY_STROBE Scan Rate = 2KHz (FSLOW /16)	
$15	SLP	W	[7:0]		Write 0x5A to sleep	xx
$16	WDGC	W	[7:0]		Write 0xA5 to clear watchdog timer	xx
$17	OPMD	W	[2]	0	Write 0 to clear RTC counter (read as high)	x
		R/W	[3]	1/0	RTC Enable/Disable	Enable
			[6:4]	000	$F_{FAST} = F_{FAOS}/1$	F_{FAOS} /1
				001	$F_{FAST} = F_{FAOS}/2$	
				010	$F_{FAST} = F_{FAOS}/4$	
				011	$F_{FAST} = F_{FAOS}/8$	
				100	$F_{FAST} = F_{FAOS}/16$	
				101	$F_{FAST} = F_{FAOS}/32$	
				110	$F_{FAST} = F_{FAOS}/64$	
				111	$F_{FAST} = F_{FAOS}/128$	
			[7]	1/0	$F_{CPU} = F_{FAST}/F_{SLOW}$ (32KHz)	F_{FAST}
$18	ELFQ	R/W	[2:0]	000	ELP frequency = BT[9] (F_{FAOS} /1024)	BT[7]
				001	ELP frequency = BT[8] (F_{FAOS} /512)	
				010	ELP frequency = BT[7] (F_{FAOS} /256)	
				011	ELP frequency = BT[6] (F_{FAOS} /128)	
				100	ELP frequency = BT[5] (F_{FAOS} /64)	
				101	ELP frequency = BT[4] (F_{FAOS} /32)	
				110	ELP frequency = BT[3] (F_{FAOS} /16)	
				111	ELP frequency = BT[2] (F_{FAOS} /8)	
			[4:3]	00	ELC frequency = RT[7] (128Hz, F_{SLOW} /256)	RT[5]
				01	ELC frequency = RT[6] (256Hz, F_{SLOW} /128)	
				10	ELC frequency = RT[5] (512Hz, F_{SLOW} /64)	
				11	ELC frequency = RT[4] (1KHz, F_{SLOW} /32)	
$19	ELC	R/W	[2:0]	000	ELP duty = 1/8	7/8
				001	ELP duty = 2/8	
				010	ELP duty = 3/8	
				011	ELP duty = 4/8	
				100	ELP duty = 5/8	
				101	ELP duty = 6/8	
				110	ELP duty = 7/8	
				111	ELP always HIGH	
			[5:3]	000	ELC duty = 1/8	1/8
				001	ELC duty = 2/8	
				010	ELC duty = 3/8	
				011	ELC duty = 4/8	
				100	ELC duty = 5/8	

Addr.	Name	R/W	Bit	Data	Description	Default
				101	ELC duty = 6/8	
				110	ELC duty = 7/8	
				111	ELC always HIGH	
			[7]	1/0	EL ON/OFF	OFF
$1A	LCDPC	R/W	[2:0]	000	Charge pump clock = RT[6] (256Hz, F_{SLOW}/128)	32KHz
				001	Charge pump clock = RT[5] (512Hz, F_{SLOW}/64)	
				010	Charge pump clock = RT[4] (1KHz, F_{SLOW}/32)	
				011	Charge pump clock = RT[3] (2KHz, F_{SLOW}/16)	
				100	Charge pump clock = RT[2] (4KHz, F_{SLOW}/8)	
				101	Charge pump clock = RT[1] (8KHz, F_{SLOW}/4)	
				110	Charge pump clock = RT[0] (16KHz, F_{SLOW}/2)	
				111	Charge pump clock = 32KHz (F_{SLOW})	
			[3]	1/0	LCD Power (Charge Pump) Enable/Disable	Enable
			[4]	1/0	Internal Voltage Regulator Enable/Disable	Enable
$1B	LCDC[2]	R/W	[2:0]	100	LCD clock = RT[6] (256Hz, F_{SLOW}/128)	RT[6]
				101	LCD clock = RT[5] (512Hz, F_{SLOW}/64)	
				110	LCD clock = RT[4] (1KHz, F_{SLOW}/32)	
				111	LCD clock = RT[3] (2KHz, F_{SLOW}/16)	
			[4:3]	00	LCD OFF	OFF
				01	LCD ON	
				10	LCD all '0'	
				11	LCD all '1'	
$1C	LVD	R/W	[4]	1/0	LVD enable/disable	Disable
		R	[7]	1/0	LVD status,VDD<LVD level / VDD>LVD level	X
$1E	SPIMD	R/W	[0]	1/0	SPI shift at Mode3/Mode0	Mode3
		R	[7]	1/0	SPI shift Processing/Done	Done
$1F	SPID	R	[7:0]		Read the shifted-in data from SDI	xx
		W	[7:0]		Latch the data in shift register and starts to shift	xx
$29	AUD	R/W	[0]	1/0	Audio Output Enable/Disable	Disable
$2B	MIX	R/W	[4]	0	CH01 = CH0 + CH1	0
				1	CH01 = CH0 + CH0	
$2F	IRC	R/W	[0]	1/0	IR data output register	H/L (by option)
			[1]	1/0	CMOS/Open-Drain of IR output	CMOS
			[2]	1/0	IR Enable/Disable	Disable
		W	[3]	0	Write 0 to initial IR counter (read as high)	x
		R/W	[5:4]	00	RFC Disable	Disable
				01	RFC output the reverse signal of CX from RR	
				10	RFC output the reverse signal of CX from RT0	
				11	RFC output the reverse signal of CX from RT1	
$30	PAIO	R/W	[7:0]	1/0	1 = Input/0 = Output	FF
$31	PBIO	R/W	[7:0]	1/0	1 = Input/0 = Output	FF
$34	PA	R	[7:0]		Read input pad data/output register data	xx
		W	[7:0]	1/0	Write to input or output port register	00
$35	PB	R	[7:0]		Read input pad data/output register data	xx
		W	[7:0]	1/0	Write to input or output port register	00

Addr.	Name	R/W	Bit	Data	Description	Default
$38	PAC	R/W	[7:0]	1/0	Pull-Low Resistor Enable/Disable of input; CMOS/Open-Drain of output	00
$39	PBC	R/W	[7:0]	1/0	Pull-Low Resistor Enable/Disable of input; CMOS/Open-Drain of output	00

7.2 RAM

The static RAM (SRAM) is organized with 128 bytes for zero page register (ZP) and stack register (STK). The ZP and STK are created to store data and save the program counter (PC) for returning. Their addressing region are $0080~$00FF and $0180~$01FF, respectively. If the address is not existed, the data will be read as unknown.

For LCD application, DPRAM and ranges from $200 to $22B (refer to chapter 8.2), and the size of DPRAM is 40B. The data for display can be updated at any time, and depicted on the LCD panel immediately.

7.3 ROM

In the NY8LP10A IC, a large program/data single ROM is provided, and the size of ROM is 16KB. Because of system information comprised in the reserved region, users are not allowed to access this part. Otherwise, it would cause a reset procedure or other unpredictable impact.

8. LCD & LED Control

The NY8LP10A IC provides users to drive LCD panel to match the most common-used panels. With the help of Bipolar Junction Transistor (BJT), the NY8LP10A IC also provides users to drive Light Emitting Diode (LED) through driver mode or sink mode. This chapter is mainly to state the power supply for LCD & LED and how it displays on the panel with the corresponding setting.

8.1 LCD Power Supply

According to LCD display theory, it is requested to use the multiple voltage levels to support LCD panel display, otherwise the display will turn to white or black permanently with single voltage level. The NY8LP10A IC contains 1/2, 1/3 bias settings for the most LCD panels. For instance, 1/3 bias needs 4 voltage levels, VSS, V1 (1/3*VLCD), V2 (2/3*VLCD), VLCD. The following descriptions, charge pump mode is configured for this application. And charge pump mode can be based on the power source (VDD), or internal voltage regulator (Vreg) by option.

8.1.1 Power Pumping Mode

For power pumping mode, it also generates the necessary voltage levels for driving the LCD panel with a different bias such as 1/2 or 1/3. Its power sources involve VDD and internal Vreg. Basically, the power source (VDD), provides the voltage to VLCD, V2, or V1, which is chosen by option.

On the contrary, internal voltage regulator only acts as the power source to V1. For internal Vreg (= 1V + 0.167V* LCDPC[6:5]), users can select 1~4 levels from 1V to 1.50V.

In charge pump mode, if the display of LCD panel is correct but weak, the method to develop this situation is to increase the clock frequency for charge pump. The relative setting for charge pump clock is defined as below. Please refer to chapter 9.3 for the LCD bias connection diagram.

Addr.	Name	R/W	Bit	Data	Description	Default
$1A	LCDPC	R/W	[2:0]	000	Charge pump clock = RT[6] (256Hz, F_{SLOW}/128)	32KHz
				001	Charge pump clock = RT[5] (512Hz, F_{SLOW}/64)	
				010	Charge pump clock = RT[4] (1KHz, F_{SLOW}/32)	
				011	Charge pump clock = RT[3] (2KHz, F_{SLOW}/16)	
				100	Charge pump clock = RT[2] (4KHz, F_{SLOW}/8)	
				101	Charge pump clock = RT[1] (8KHz, F_{SLOW}/4)	
				110	Charge pump clock = RT[0] (16KHz, F_{SLOW}/2)	
				111	Charge pump clock = 32KHz (F_{SLOW})	
			[3]	1/0	LCD Power (Charge Pump) Enable/Disable	Enable
			[4]	1/0	Internal Voltage Regulator Enable/Disable	Enable
			[6:5][1]	00	Voltage Regulator (Vreg) = 1V	1.50V
				01	Voltage Regulator (Vreg) = 1.17V	
				10	Voltage Regulator (Vreg) = 1.33V	
				11	Voltage Regulator (Vreg) = 1.50V	

For the LCD display power being stable, the procedure of LCD turning on and turning off is suggested below.

```
M_LCD_Regulator_ON      ;turn on regulator
M_LCD_Charge_ON         ;turn on charge pump
M_LCD_ON                ;turn on LCD
…
M_LCD_OFF               ;turn off LCD
M_LCD_Charge_OFF        ;turn off charge pump
M_LCD_Regulator_OFF     ;turn off regulator
```

8.2 LCD & LED RAM Alignment

The following table is used to store the data for the LCD or LED display, 6 common pins, and the maximum address is $022B for DPRAM. It is strongly recommended to initialize the data of DPRAM. Because of uncertain data stored in undecided address, it may cause the defect of display.

	SEG[7:0]	SEG[15:8]	SEG[23:16]	SEG[31:24]	
	(bit[7:0])	(bit[7:0])	(bit[7:0])	(bit[7:0])	(bit[7:0])
COM0	$200H	$201H	$202H	$203H	$204H
COM1	$208H	$209H	$20AH	$20BH	$20CH
COM2	$210H	$211H	$212H	$213H	$214H
COM3	$218H	$219H	$21AH	$21BH	$21CH
COM4	$220H	$221H	$222H	$223H	$224H
COM5	$228H	$229H	$22AH	$22BH	$22CH
COM6	$230H	$231H	$232H	$233H	$234H
COM7	$238H	$239H	$23AH	$23BH	$23CH

8.3 LCD Display System

As for a plenty of LCD display usages, the NY8LP10A IC provides three mask options to match the most common-used LCD panels of the market, they are:

LCD Duty: 1/(X) duty. (X = 2 ~6)
LCD Bias: 1/(Y) bias. (Y = 2 ~ 3)
LCD SEG: (Z) segments. (Z = 0 ~ 31)

Because of the variety of panels, each display system should set its own combination from those three options. Here is an example works under 1/4 duty, 1/3 bias and 8 segments, the mask options should set X as 4, Y as 3 and Z as 8 to match the specific LCD panel, shown in Figure 5-1. According to Table 1, if users want to display digits such as “1.2.3.4.”, the relative data has to be written in DPRAM to show out. Those data would be 0x7D, 0xA6, 0xE8 and 0xBE and LCD Panel will display “1.2.3.4.” when LCD is turned on by instruction.

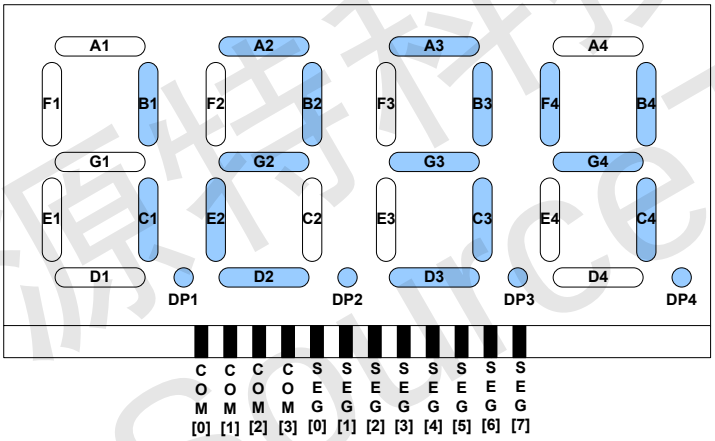

Figure 8-1: 4 digits LCD Panel

Addr.	Item	SEG[7]	SEG[6]	SEG[5]	SEG[4]	SEG[3]	SEG[2]	SEG[1]	SEG[0]
$200	COM[0]	D4	DP4	D3	DP3	D2	DP2	D1	DP1
$208	COM[1]	C4	E4	C3	E3	C2	E2	C1	E1
$210	COM[2]	G4	F4	G3	F3	G2	F2	G1	F1
$218	COM[3]	B4	A4	B3	A3	B2	A2	B1	A1

Table 1: LCD panel mapping

The frame rate for each display system is based on the setting of duty and LCD clock, and these can be selected by setting special function register (SFR). Users can select through setting of SFR, 64Hz, 128Hz, 256Hz, 512Hz, 1KHz or 2KHz for the LCD display (All of the LCD frame rates are sourced from slow clock F_{SLOW}, 32KHz). The following is the formula to calculate frame rate:

Frame Rate = LCD Clock/COM Number

If the pattern on the LCD panel starts to flash, it is suggested to tune to higher rate to be correspondent with the desired display.

Addr.	Name	R/W	Bit	Data	Description	Default
$1B	LCDC	R/W	[2:0]	0x0	LCD clock = RT[8] (64Hz, F_{SLOW}/512)	RT[6]
				0x1	LCD clock = RT[7] (128Hz, F_{SLOW}/256)	
				100	LCD clock = RT[6] (256Hz, F_{SLOW}/128)	
				101	LCD clock = RT[5] (512Hz, F_{SLOW}/64)	
				110	LCD clock = RT[4] (1KHz, F_{SLOW}/32)	
				111	LCD clock = RT[3] (2KHz, F_{SLOW}/16)	
			[4:3]	00	LCD OFF	OFF
				01	LCD ON	
				10	LCD all '0'	
				11	LCD all '1'	

For the most common-used LCD panels, the relationship between LCD duty and LCD bias is shown as below table.

LCD bias ≈ $1/(1+\sqrt{duty})$	
LCD duty	**LCD bias select**
2	1/2
3	1/2, 1/3
4	1/2, 1/3
5	1/2, 1/3
6	1/3,

Either common pins or segment pins are operated under alternative voltage level according to the mode it acts. The following lists voltage level of corresponding bias settings and users have to connect with the identical power system.

9. LCD WAVEFORMS

The following lists voltage level of corresponding bias settings and users have to connect with the identical power system.

Bias	Voltage Level
1/2	VSS, V1 (=1/2*VLCD), VLCD
1/3	VSS, V1 (=1/3*VLCD), V2 (=2/3*VLCD), VLCD

The LCD timing waveforms are shown as Fig.6-1 ~ Fig.6-2.

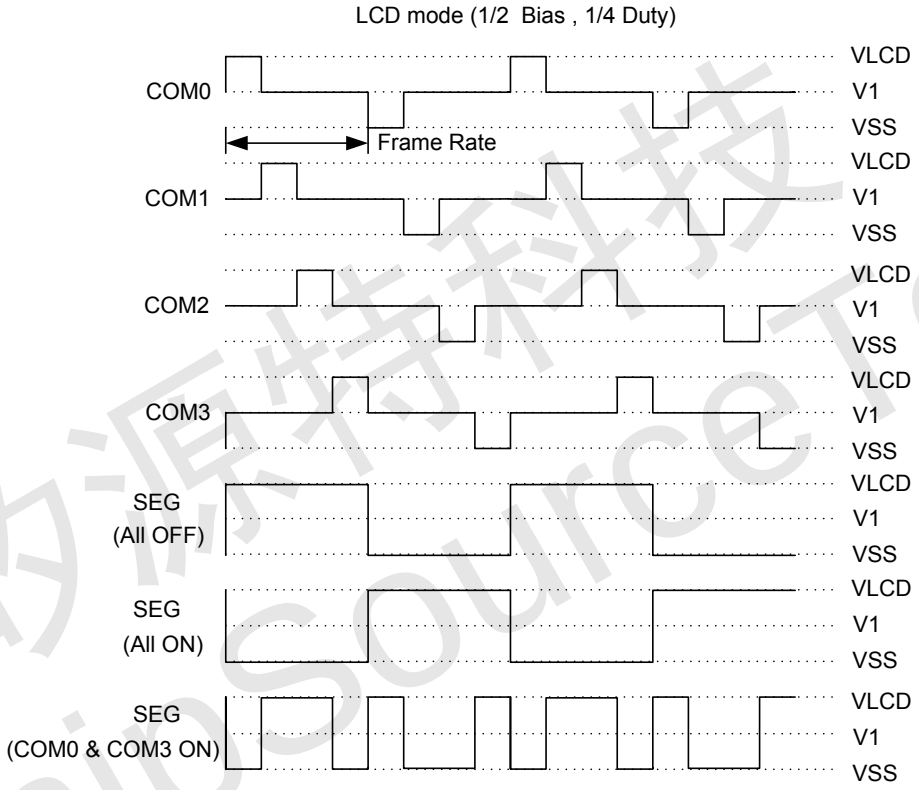

Fig.9-1: LCD timing waveform of 1/2 bias, 1/4 Duty

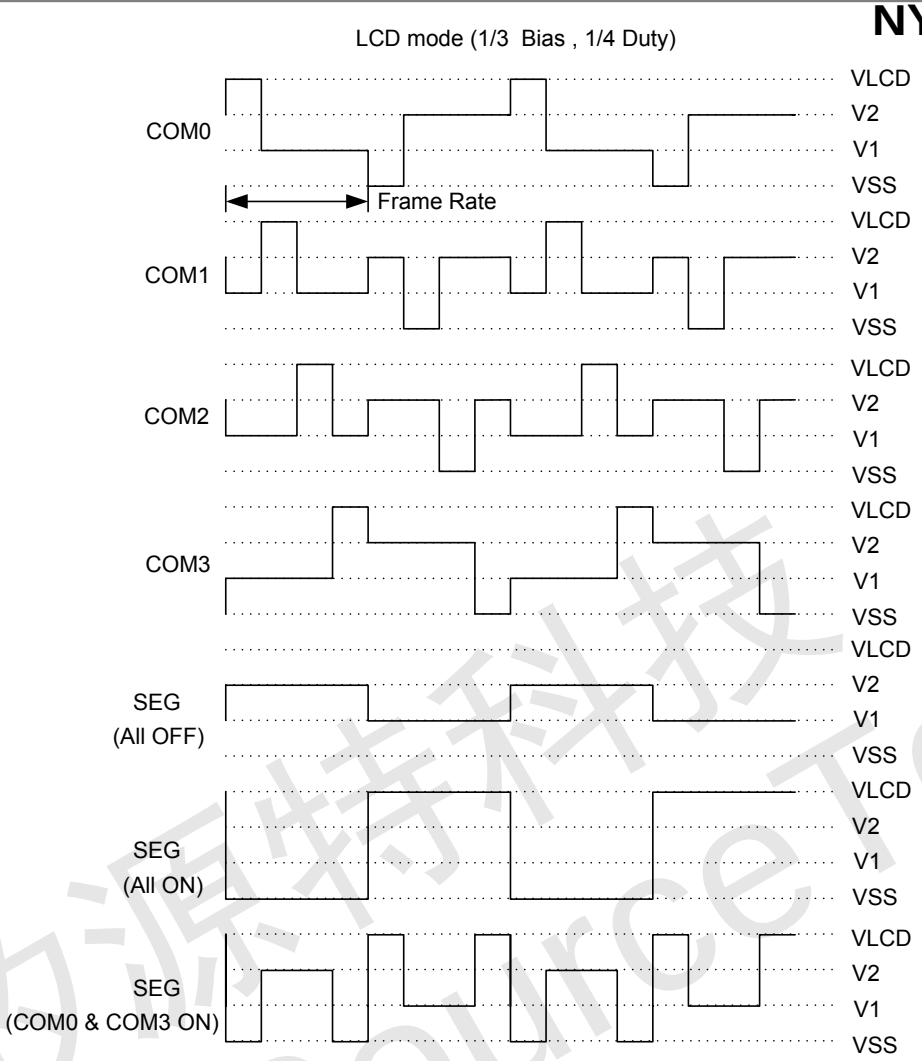

Fig.9-2: LCD timing waveform of 1/3 bias, 1/4 Duty

9.1 LED Display System

With the help of Bipolar Junction Transistor (BJT), the NY8LP10A IC also provides users to drive Light Emitting Diode (LED) through driver mode or sink mode. LCD and LED functions must all be enabled with the corresponding setting. LED power source is based on VDD without bias circuitry. And users should initialize the data of DPRAM, and choose the frame rate for each display system, which is the same as LCD display by setting control register (LCD).

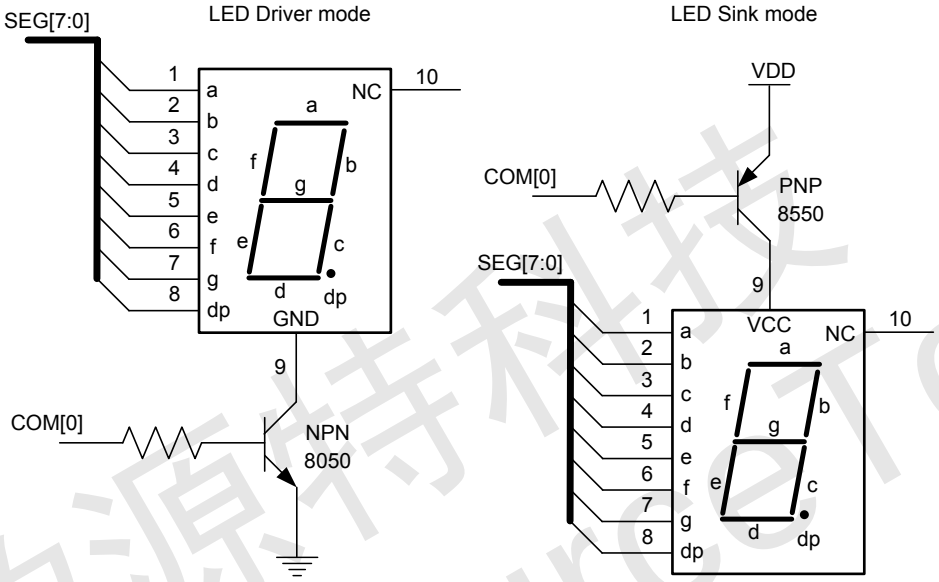

Figure 9-3: LED driving circuitry of drive mode & sink mode

The LED timing waveforms are shown as the following Figure 5-6 & Figure 5-7.

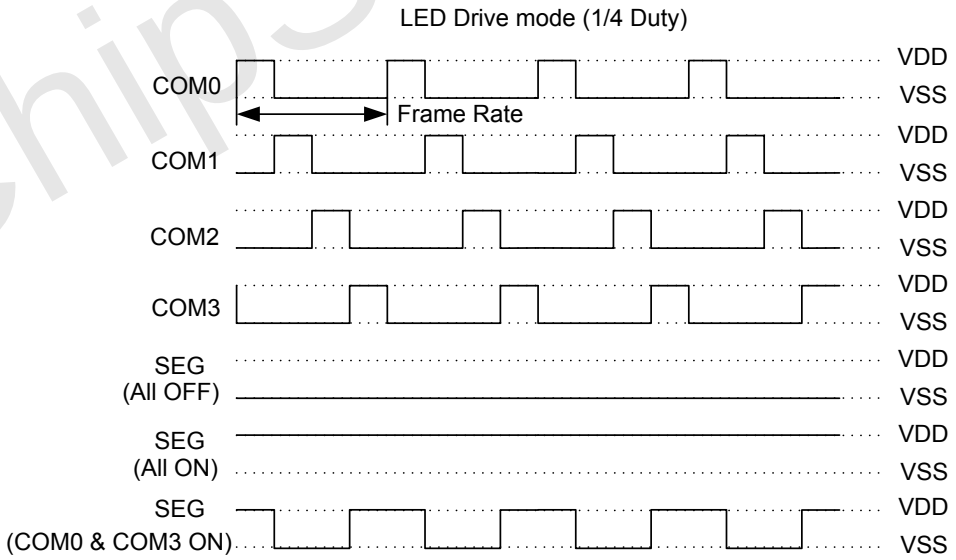

Figure 9-4: LED timing waveform of driver mode

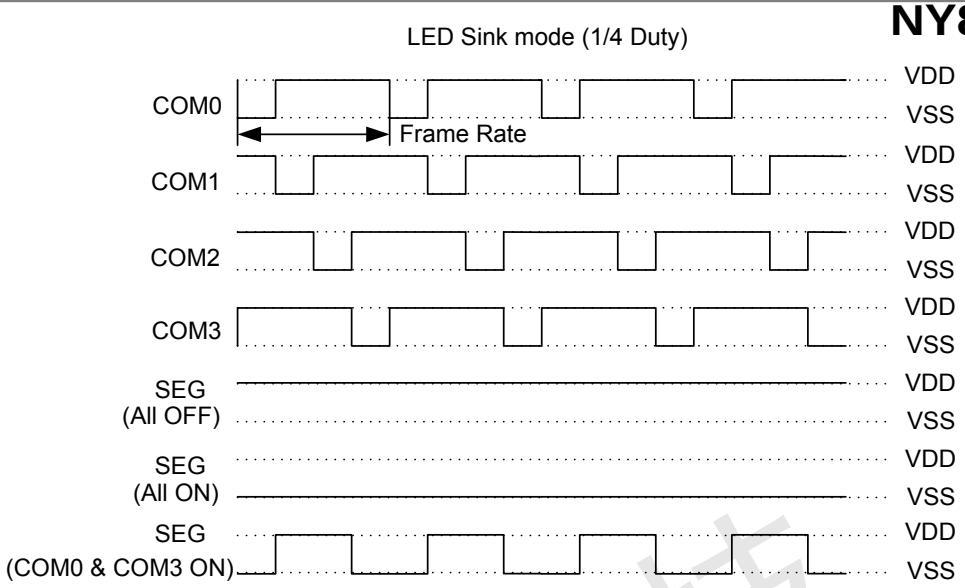

Figure 9-5: LED timing waveform of sink mode

9.1.1 TMxD (x=0, 1, 2)

The TMxD registers include a set 8-bit timer reload value latch and a set 8-bit downward counter of the `x' channel. With the data loaded in the latch, the counter counts down until to 0. If the register is set to the reload mode, the counter will be automatically reloaded from the latch, and the reload period is TMxD+1 (TMxD ≠0). When the counter counts to 0, the audio engine plays the audio data.

$TMxD = (F_{TCS} / F_{SR}) - 1$

TMxD: Timer value in decimal

F_{TCS}: Frequency of the timer clock source

F_{SR}: Frequency of the sample rate

In theory, there are TMxD+1 (TMxD ≠0) cycles of timer used to control an accurate period of time. ***But actually the maximum deviation is perhaps 1 cycle of timer, because timer enable signal is asynchronous to timer clock source. And it is strongly recommended to speed up timer clock source to decrease the deviation.*** Users can read back the content of counter through register TMxD by instruction.

Addr.	Name	R/W	Bit	Data	Description	Default
$00	TM0D	R	[7:0]		Read the Timer0 counting data[7:0]	xx
		W	[7:0]		Preload Timer0 data[7:0]	xx
$04	TM1D	R	[7:0]		Read the Timer1 counting data[7:0]	xx
		W	[7:0]		Preload Timer1 data[7:0]	xx
$08	TM2D	R	[7:0]		Read the Timer2 counting data[7:0]	xx
		W	[7:0]		Preload Timer2 data[7:0]	xx

9.1.2 TMxC (x=0, 1, 2)

The TMxC registers indicate the TMx clock source of the `x' channel. Different channel mode has different frequency of the TMxC. The value of the TMxC affects the tone. Besides, real timer (RT[n]) is based on F_{SLOW} (32.768KHz), and base timer (BT[n]) is divided from F_{FAOS} (4MHz), which is selected by option. So BT[n] will be slow down as well as the fast system clock. The settings of TMxC are shown as below.

Addr.	Name	R/W	Bit	Data	Description	Default
$01	TM0C	R/W	[3:0]	000x	Timer0 clock = CX	BT[0]
				001x	Timer0 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer0 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer0 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer0 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer0 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer0 clock = BT[6] (F_{FAOS}/128)	
				1001	Timer0 clock = BT[5] (F_{FAOS}/64)	
				1010	Timer0 clock = BT[4] (F_{FAOS}/32)	
				1011	Timer0 clock = BT[3] (F_{FAOS}/16)	
				1100	Timer0 clock = BT[2] (F_{FAOS}/8)	
				1101	Timer0 clock = BT[1] (F_{FAOS}/4)	
				1110	Timer0 clock = BT[0] (F_{FAOS}/2)	
				1111	Timer0 clock = F_{FAOS}	
			[4]	1/0	Timer0 Reload/One shot	1
			[5]	1/0	Tone0 Enable/Disable	Enable
			[7:6]	00	Timer0 clock stop mode OFF	OFF
				01	Timer0 clock stopped by Timer2 overflow	
				10	Timer0 clock stopped by a full cycle of CX	
				11	Timer0 clock stopped by a full cycle of Timer2 clock	
$05	TM1C	R/W	[3:0]	000x	Timer1 clock = TM0D[7]	BT[0]
				001x	Timer1 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer1 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer1 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer1 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer1 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer1 clock = BT[6] (F_{FAOS}/128)	
				1001	Timer1 clock = BT[5] (F_{FAOS}/64)	
				1010	Timer1 clock = BT[4] (F_{FAOS}/32)	
				1011	Timer1 clock = BT[3] (F_{FAOS}/16)	
				1100	Timer1 clock = BT[2] (F_{FAOS}/8)	
				1101	Timer1 clock = BT[1] (F_{FAOS}/4)	
				1110	Timer1 clock = BT[0] (F_{FAOS}/2)	
				1111	Timer1 clock = F_{FAOS}	
			[4]	1/0	Timer1 Reload/One shot	1
$09	TM2C	R/W	[3:0]	000x	Timer2 clock = TM1D[7]	BT[2]
				001x	Timer2 clock = RT[13] (2Hz, F_{SLOW}/16384)	
				0100	Timer2 clock = RT[11] (8Hz, F_{SLOW}/4096)	
				0101	Timer2 clock = RT[9] (32Hz, F_{SLOW}/1024)	
				0110	Timer2 clock = RT[7] (128Hz, F_{SLOW}/256)	
				0111	Timer2 clock = RT[5] (512Hz, F_{SLOW}/64)	
				1000	Timer2 clock = BT[8] (F_{FAOS}/512)	

Addr.	Name	R/W	Bit	Data	Description	Default
				1001	Timer2 clock = BT[7] (F_{FAOS}/256)	
				1010	Timer2 clock = BT[6] (F_{FAOS}/128)	
				1011	Timer2 clock = BT[5] (F_{FAOS}/64)	
				1100	Timer2 clock = BT[4] (F_{FAOS}/32)	
				1101	Timer2 clock = BT[3] (F_{FAOS}/16)	
				1110	Timer2 clock = BT[2] (F_{FAOS}/8)	
				1111	Timer2 clock = BT[1] (F_{FAOS}/4)	
			[4]	1/0	Timer2 Reload/One shot	1

9.1.3 TMxEN(x=0, 1, 2)

The bit0 of TM0EN, TM1EN and TM2EN are mainly to control these timers to turn on/off respectively. As the value is 1, the timer is turned on and the counter starts counting downward.

Addr.	Name	R/W	Bit	Data	Description	Default
$02	TM0EN	R/W	[0]	1/0	Timer0 Enable/Disable	Disable
$06	TM1EN	R/W	[0]	1/0	Timer1 Enable/Disable	Disable
$0A	TM2EN	R/W	[0]	1/0	Timer2 Enable/Disable	Disable

9.2 Buzzer Control

The NY8LP10A also provides buzzer output, and users can select single or dual channel buzzer, which is controlled by MIX[4]. When MIX[4]=1, single buzzer toggled by Timer0 overflow, otherwise when MIX[4]=0, dual buzzer toggled by Timer0 or Timer1 overflow, and the buzzer switching rate is equal to F_{SLOW}. The diagram of buzzer module is shown as Figure 6-3.

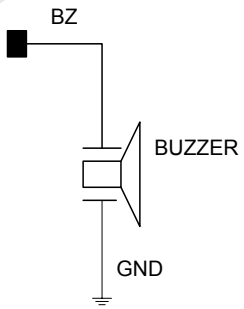

Figure 9-6: The diagram of buzzer module

10. I/O Control

10.1 I/O Ports

There are 14 I/O ports (by body and option), designated as PAy through PBy, and y=0~7, the structure of I/O ports is shown as Figure 7-1, and users can also enable them by setting options. The bi-direction I/O port can be an input or output by the value of control register PXIO (X = A/B). If the register value is 1, the port will be set as an input; therefore, the value 0 means output setting. Users can set the control register PXC (X = A/B) to define the I/O ports to be with/without a pull-low resister if input or configured as COMS/Open-Drain type if output. As for the internal pull-low resistor of input, it can be set as strong or weak pull-low through option. The weak one is about 1MΩ@3V for normal application and the strong one is about 100KΩ@3V.

If Port X (X = A/B) is key change wake-up source, users must read input pads data and write the value to input port registers before enter Standby/Halt mode. The system will be waked up as long as one of input pads status change.

If Port A is key change wake-up source, the code of entering Standby mode is shown below.

```
LDA  OPMD      ;
ORA  #$08      ;
STA  OPMD      ; Set OPMD[3] as high
LDA  #$FF
STA  PAIO      ; Set PortA as Input
LDA  PA        ; Read PortA data
STA  PA        ; Write to PortA register

LDA  #$5A
STA  SLP       ; Enter Standby mode
```

And the code of entering Halt mode is shown below.

```
LDA  OPMD      ;
AND  #$F7      ;
STA  OPMD      ; Clear OPMD[3] to low
LDA  #$FF
STA  PAIO      ; Set PortA as Input
LDA  PA        ; Read PortA data
STA  PA        ; Write to PortA register

LDA  #$5A
STA  SLP       ; Enter Halt mode
```

Input/Output port : PX0~7(X=A/B)

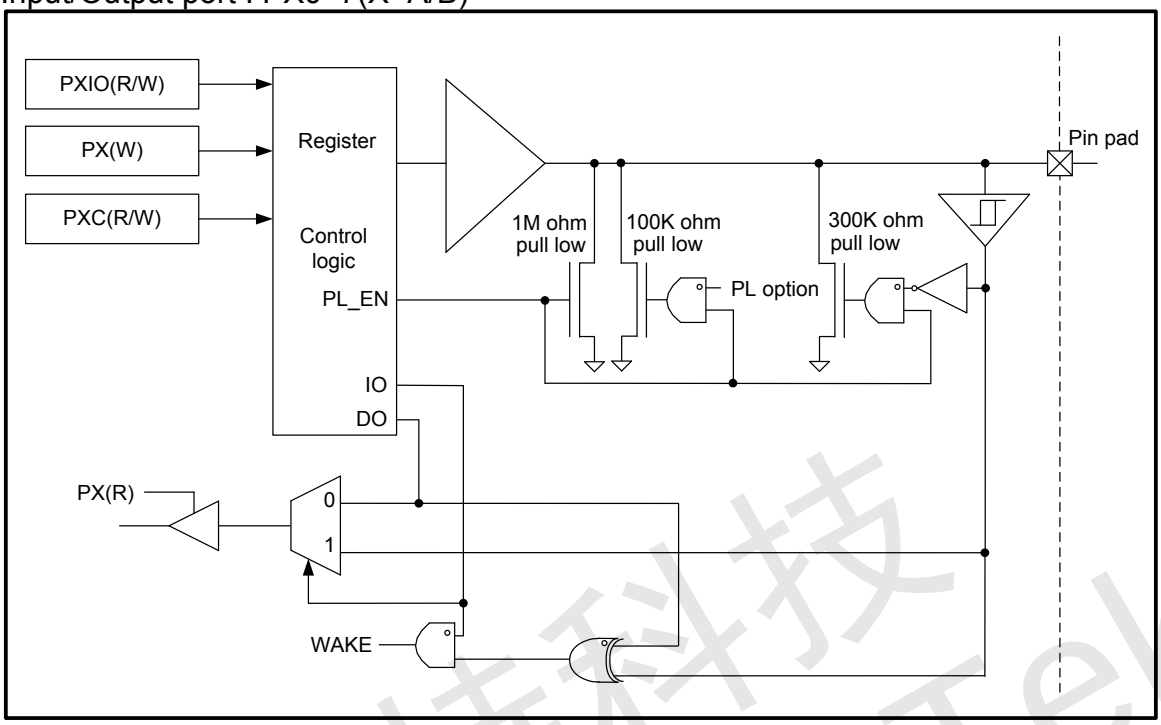

Figure 10-1: I/O port Configuration

The following tables describe the functionalities of each register:

Addr.	Name	R/W	Bit	Data	Description	Default
$30	PAIO	R/W	[7:0]	1/0	1 = Input/0 = Output	FF
$31	PBIO	R/W	[7:0]	1/0	1 = Input/0 = Output	FF
$34	PA	R	[7:0]		Read input pad data/output register data	xx
		W	[7:0]	1/0	Write to input or output port register	00
$35	PB	R	[7:0]		Read input pad data/output register data	xx
		W	[7:0]	1/0	Write to input or output port register	00
$38	PAC	R/W	[7:0]	1/0	Pull-Low Resistor Enable/Disable of input; CMOS/Open-Drain of output	00
$39	PBC	R/W	[7:0]	1/0	Pull-Low Resistor Enable/Disable of input; CMOS/Open-Drain of output	00

10.2 Matrix Key Strobe

The NY8LP10A utilizes the partial timing of the LCD display to scan the key strobe circuitry. These scanning output pins are SEG7~SEG22 (K0~15). The input part of key strobe circuitry (KI0~3) comprises PB I/O ports. With the control register KSB, users can define the scan rate, the interrupt mode and the key strobe output, and see as below table.

Addr.	Name	R/W	Bit	Data	Description	Default
$14	KSB	R/W	[3:0]		Key strobe output segment select	0000
			[4]	1/0	All/One segments selected as key strobe output	One
			[5]	0	Key strobe interrupt = key strobe occurs	0
				1	Key strobe interrupt = each key strobe scanning cycle	
			[7:6]	00	Key strobe scan rate = RT[6] (256Hz, F_{SLOW}/128)	RT[6]
				01	Key strobe scan rate = RT[5] (512Hz, F_{SLOW}/64)	
				10	Key strobe scan rate = RT[4] (1KHz, F_{SLOW}/32)	
				11	Key strobe scan rate = RT[3] (2KHz, F_{SLOW}/16)	

In the Figure 10-2, the key strobe scanning inputs (KI0~3) are connected with PB0~7 (by body), and key strobe outputs (K0~15) are connected with LCD panel SEG7~22 (by body). When any of the buttons is pressed, a HIGH signal will deliver from the segment pins to port side and be detected by program to execute the additional instructions. In this case, if the interrupt enable (IEF[7]) is provided, the interrupt is accepted.

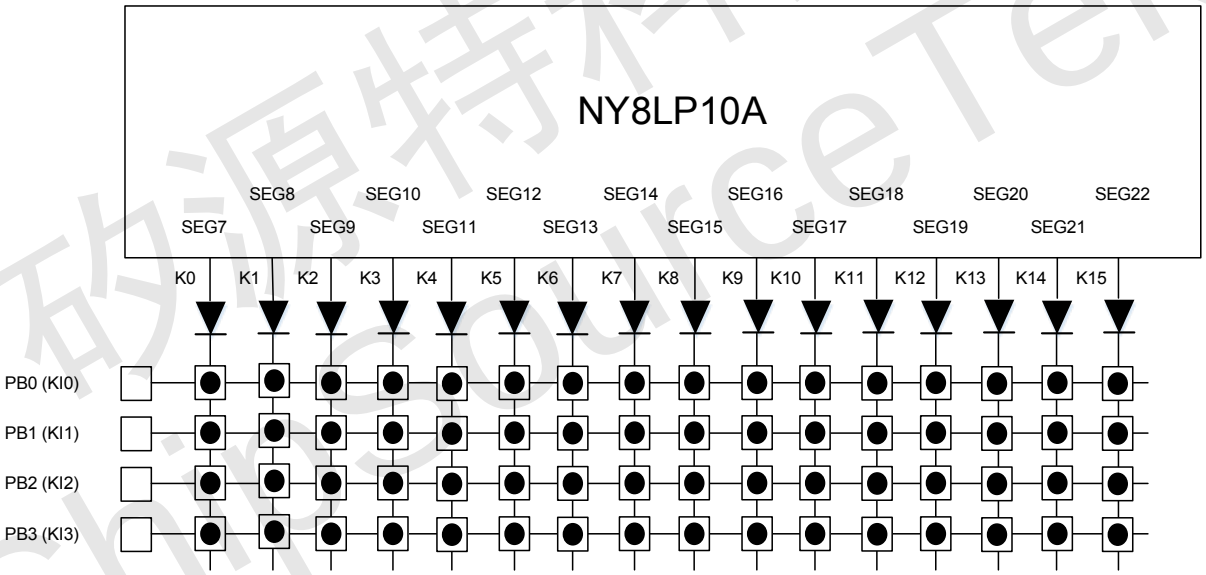

Figure 10-2: The 4x16 key strobe

The procedure of key strobe application is shown below.

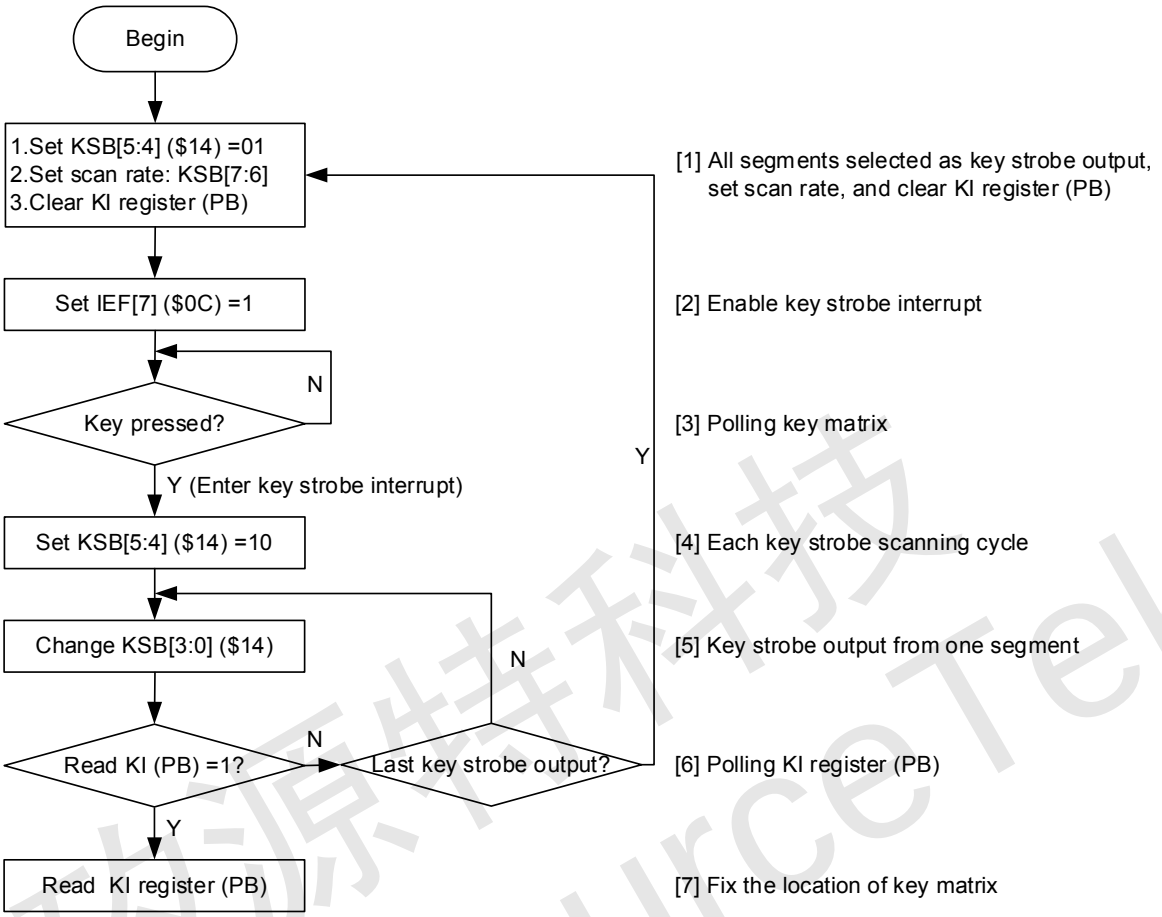

Figure 10-3: Flow chart of the key strobe scanning

Besides, the system can also be waked up from Standby mode by key strobe wake-up source, which is similar as key change. Before enter Standby mode, users must set the KI port as input, and clear KI register through writing 0 to PB. Normally key strobe outputs are controlled by KSB[3:0] ($14), when any of the buttons is pressed, a HIGH signal will deliver from the segment pins to port side, which will wake up the system.

Addr.	Name	R/W	Bit	Data	Description	Default
$35	PB	R	[7:0]		KSB: Read register data	xx
		W	[7:0]	1/0	KSB: Write 0 to Clear register data	00

For example, if PB is KSB wake-up source, the code of entering Standby mode is shown below.

```
LDA  OPMD      ;
ORA  #$08      ;
STA  OPMD      ; Set OPMD[3] as high
LDA  #$FF
STA  PBIO      ; Set PB as input
LDA  #$00
```

```
STA  PB          ; Clear PB data register

LDA  #$5A
STA  SLP         ; Enter Standby mode
```

On the contrary, for avoiding awaking Halt mode wrongly, the code of entering Halt mode is shown below.

```
LDA  OPMD        ;
AND  #$F7        ;
STA  OPMD        ; Clear OPMD[3] to low
LDA  #$00
STA  PBIO        ; Set PB as output
LDA  #$00
STA  PB          ; Clear PB data register

LDA  #$5A
STA  SLP         ; Enter Halt mode
```

11. Other Applications (by option)

The NY8LP10A IC supports many applications with external components, such as EL SPI, RFC, and IR. The continued sections will describe their functionalities and operations.

11.1 Electroluminescent (EL) Back Light Driver

NY8LP10A provides an EL panel driver for the back light of the LCD panel. The typical EL driver circuitry consists of built-in EL driver and external diode, transistor, resistor and inductance. The circuitry and waveform are shown as Figure 11-1 and Figure 11-2. Users can choose different voltage pump frequency, duty cycle and Enable/Disable frequency through register ELFQ and ELC to operate. To enable the EL block, users still have to enable the relative option.

EL pump (ELP) frequency is divided from fast oscillator clock (4MHz/2MHz/500KHz), so it will be slow down as well as F_{FAOS}. EL clear (ELC) frequency is based on slow oscillator clock (32.768KHz). With different frequency and duty for ELP and ELC, the voltage of EL panel can be generated up to 100V or above.

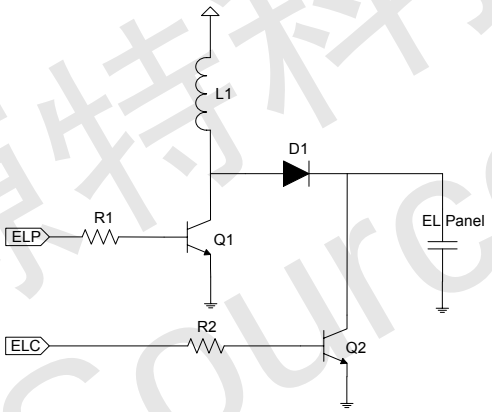

Figure 11-1: The circuitry of EL Driver.

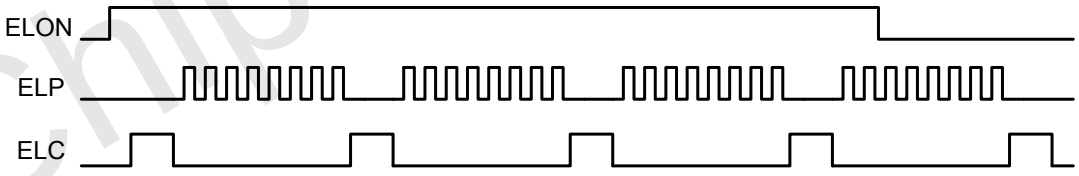

Figure 11-2: The timing waveform of ELP & ELC.

The procedure of EL application goes to three major parts. First, once ELON is enabled, the first pulse of ELC is to discharge uncertain charge stored in EL panel before pumping. Then, the ELP pin will output clocks to pump voltage to the EL panel and the ELC pin will output the pulse right after the last ELP pulse to discharge the EL panel. Eventually, when ELON signal is disabled, the ELC pin will output a pulse to discharge the EL panel after the last pump clock. It insures that there is no residual voltage to cause damage. The relative settings for frequencies and duties are set by the control registers and shown as the following tables.

Addr.	Name	R/W	Bit	Data	Description	Default
$18	ELFQ	R/W	[2:0]	000	ELP frequency = BT[9] (F_{FAOS}/1024)	BT[7]
				001	ELP frequency = BT[8] (F_{FAOS}/512)	
				010	ELP frequency = BT[7] (F_{FAOS}/256)	
				011	ELP frequency = BT[6] (F_{FAOS}/128)	
				100	ELP frequency = BT[5] (F_{FAOS}/64)	
				101	ELP frequency = BT[4] (F_{FAOS}/32)	
				110	ELP frequency = BT[3] (F_{FAOS}/16)	
				111	ELP frequency = BT[2] (F_{FAOS}/8)	
			[4:3]	00	ELC frequency = RT[7] (128Hz, F_{SLOW}/256)	RT[5]
				01	ELC frequency = RT[6] (256Hz, F_{SLOW}/128)	
				10	ELC frequency = RT[5] (512Hz, F_{SLOW}/64)	
				11	ELC frequency = RT[4] (1KHz, F_{SLOW}/32)	
$19	ELC	R/W	[2:0]	000	ELP duty = 1/8	7/8
				001	ELP duty = 2/8	
				010	ELP duty = 3/8	
				011	ELP duty = 4/8	
				100	ELP duty = 5/8	
				101	ELP duty = 6/8	
				110	ELP duty = 7/8	
				111	ELP always high	
			[5:3]	000	ELC duty = 1/8	1/8
				001	ELC duty = 2/8	
				010	ELC duty = 3/8	
				011	ELC duty = 4/8	
				100	ELC duty = 5/8	
				101	ELC duty = 6/8	
				110	ELC duty = 7/8	
				111	ELC always high	
			[7]	1/0	EL Enable/Disable	Disable

11.2 Serial Peripheral Interface (SPI)

NY8LP10A supports only master mode to access serial Flash/SRAM memory. The relative control register SPIMD and SPID are shown as the continued tables. To enable the SPI block, users still have to enable the relative option.

Addr.	Name	R/W	Bit	Data	Description	Default
$1E	SPIMD	R/W	[0]	1/0	SPI shift at Mode3/Mode0	Mode3
		R	[7]	1/0	SPI shift Processing/Done	Done
$1F	SPID	R	[7:0]		Read the shifted-in data from SDI	xx
		W	[7:0]		Latch the data in shift register and starts to shift	xx

Users can set control register SPIMD to select mode3 or mode0 for shifting data out and also read back bit7 of SPIMD to recognize the state of process for further control. SPI always shifts data at rising edge of SCK,

and when SPI enters idle mode, SCK keeps high at mode3, otherwise keep low at mode0. For WRITE procedure, it starts with data loaded in register SPID, then shifts 8-bit data out from MSB to LSB at upcoming eight rising edges, and confirms the end of process by reading bit7 of SPIMD. If the bit7 of SPIMD is high, that means system is busy and is not allowed to execute an additional byte of data until it turns to low.

For the connection with external Flash/SRAM, the applied pins are SDI, SDO and SCK. The SDI is the input pin to receive data from the external device, and the SDO is the output pin to deliver data. The SCK is the output pin to offer the clock signal, and configured as F_{CPU}. However, the enable pin for the external device can share with one of I/O ports and define it as output.

For example, the following waveform shows that SPID data is preloaded data “0xAA” and shift out “10101010” in order and shift data into the control register SPID with data “0x55”. Because the programmer knows the process is to send out the data to the external slave device, the data stored in register SPID will be ignored.

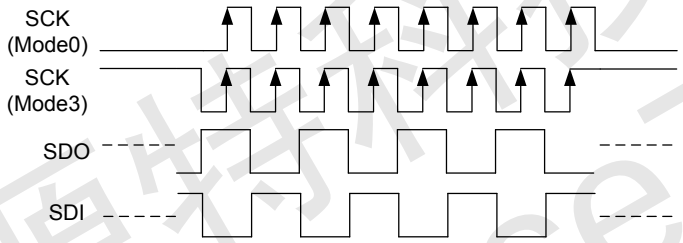

Figure 11-1: The timing waveform of SPI operating mode

11.3 Resistor to Frequency Converter (RFC)

The resistor to frequency converter (RFC) is used to compare two different sensors with the reference resistor individually. The operating principle is based on a RC oscillator network. This Figure 8-4 shows the block diagram of RFC:

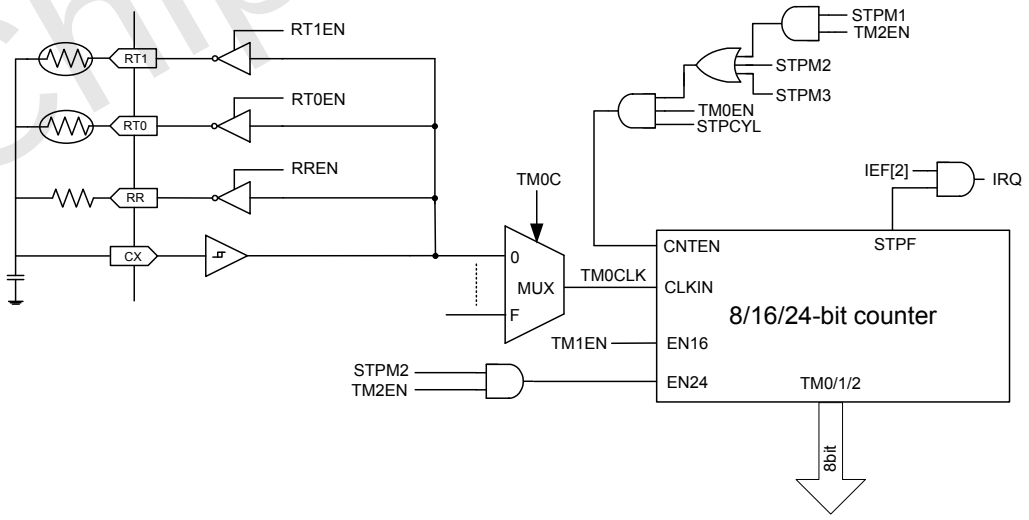

Figure 11-2: The circuitry of RFC.

The architecture of RFC contains four external pins:

CX: the oscillation Schmitt trigger input
RR: the reference resistor output pin
RT0: the temperature sensor output pin
RT1: the humidity sensor output pin (this can also be used as another temperature sensor, or can even be left floating), which can be set as I/O by option

The following definitions of RFC circuitry signals are described below.

RREN/RT0EN/RT1EN: To enable the tri-state buffer to output the reverse signal of CX.
STPCYL: To activate 8/16/24-bit counter.
STPM1/2/3: The control signal defined Timer0 stop mode.
TM0EN/TM1EN/TM2EN: The signals defined the Timer0/1/2 are turned on.
TM0CLK: The clock for Timer0, defined by register TM0C.
IEF2: The bit2 of register IEF.
STPF: The stopped signal as counter is ended to count.
EN16: The signal to extend counter from 8-bit to 16-bit.
EN24: The signal to extend counter from 16-bit to 24-bit.

11.3.1 RC Oscillation Network

The RFC circuitry may build up 3 RC oscillation networks through RR, RT0, or RT1 and CX pins with external resistors and can be disabled by mask option. The oscillation network can be built up by setting register IRC to enable RR, RT0 and RT1 respectively, but only one RC oscillation network is active simultaneously.

Addr.	Name	R/W	Bit	Data	Description	Default
$2F	IRC	R/W	[5:4]	00	RFC Disable	Disable
				01	RFC output the reverse signal of CX from RR	
				10	RFC output the reverse signal of CX from RT0	
				11	RFC output the reverse signal of CX from RT1	

As relative settings are ready, the clock will be generated by the oscillation network and feedback to the counter through CX pin. If counter is enabled, the clock will be to down-count the preloaded value of counter.

The RC oscillation network needs to set up with three simple steps:

1. Connect RR, RT0 and RT1 with a resistor respectively and a capacitor between CX and VSS. The above RFC circuitry shows the connection of these networks.
2. Switch on RREN, RT0EN or RT1EN to output the reverse signal of CX, and then it forms the clock source feedback from RC oscillation network. Those pins will be tri-state as the corresponding enable signal is off.

3. Choose Stop Mode1, Stop Mode2, or Stop Mode3 (TM0C[7:6]=01/10/11) and turn on Timer0, Timer0&1, or Timer0&1&2 to enable 8/16/24-bit counter.

It strongly recommends users to switch on output pin for each RC network before the counter is activated to get better clock signal from CX pin.

The NY8LP10A IC provides 3 counting modes for the operation of the counter. Each mode can trigger Timer0 interrupt if IEF2 is enabled, and will be described in the following sections. Users can program the control register TM0C to choose the mode it will act, and the table is listed as below. For the other settings of timers, refer to chapter 6.1.4~6.1.6.

Addr.	Name	R/W	Bit	Data	Description	Default
$01	TM0C	R/W	[7:6]	00	Timer0 clock stop mode OFF	OFF
				01	Timer0 clock stopped by Timer2 overflow	
				10	Timer0 clock stopped by a full cycle of CX	
				11	Timer0 clock stopped by a full cycle of Timer2 clock	

11.3.2 Timer0 Counting within Timer2 Overflow Cycle (STOP Mode1)

In this mode, Timer2 will dominate the operation of the Timer0(8-bit), or Timer0&1(16-bit) counter. Firstly, the value of Timer2 for counting is loaded by instruction, and the counter won't start to operate until Timer0, or Timer0&1 and Timer2 are enabled. **It strongly recommends users to switch on Timer2 at last.** And then STPCYL goes high, the Timer2 will count down (Y-2) cycles once its falling edge occurs, supposed Y is the initial value. By counting to 0x00, the Timer2 will be stopped to end the process. In this case, if the interrupt enable (IEF[2]) is provided, the interrupt is accepted.

In theory, there are Y full cycles of Timer2 used to control an accurate period of time for CX pin clocking into 8/16-bit counter. **But actually the maximum deviation is perhaps 2 cycle of Timer2**, because Timer2 enable signal is asynchronous to Timer2 clock source. And the deviation will be decreased when Timer2 clock source speed up. Users can read back the content of counter through register TMxD (x=0, 1) by instruction. The procedure of RFC counter controlled by Timer2 is shown as Figure 8-5.

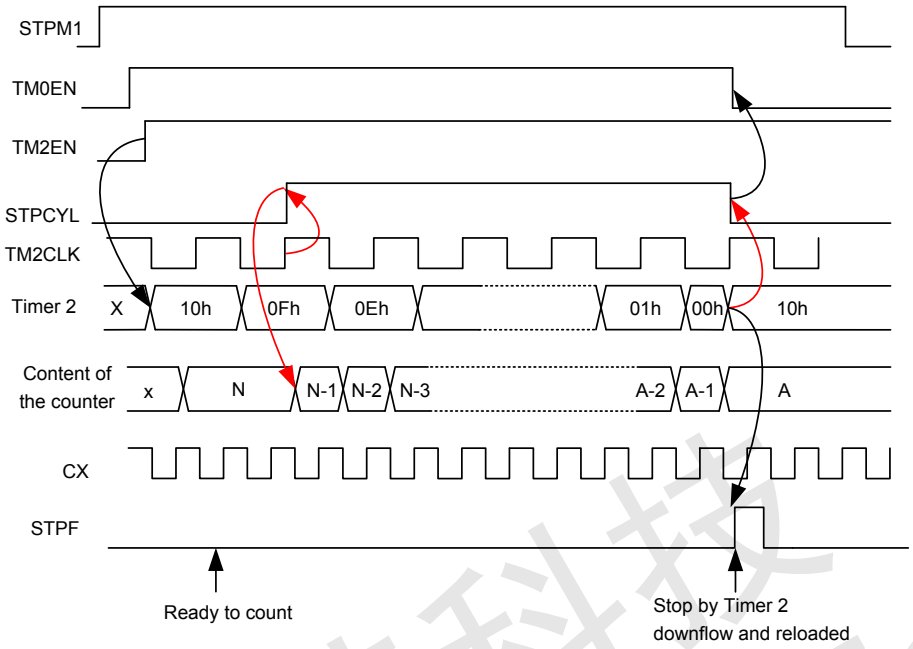

Figure 11-3: Timing of the RFC counter controlled by Timer2 downflow

11.3.3 Timer0 Counting within a Full CX Cycle (STOP Mode 2)

This is the other way to utilize the Timer0 (8-bit), Timer0&1 (16-bit), or Timer0&1&2 (24-bit) counter. Here, CX pin is used to control the enabling of the counter and the clock of Timer0 (TM0CLK), which defined by register TM0C, becomes the clock source of 8/16/24-bit counter. The procedure is quiet similar with previous mode, choose mode it will act, turn on Timer0, Timer0&1, or Timer0&1&2 and load a specific value into the counter at beginning. **It strongly recommends users to switch on Timer0 at last.**

As the first falling edge of CX clock comes out, the STPCYL will goes high to enable the counter. Then the counter will start to count downward until the second falling edge of CX clock occurs. At the time, the stop flag (STPF) will be high to disable TM0CLK and end of procedure. In this case, if the interrupt enable (IEF[2]) is provided, the interrupt is accepted.

Users can read back the content of counter through register TMxD (x=0, 1, 2) by instruction. The timing procedure of this mode is shown as Figure 8-6.

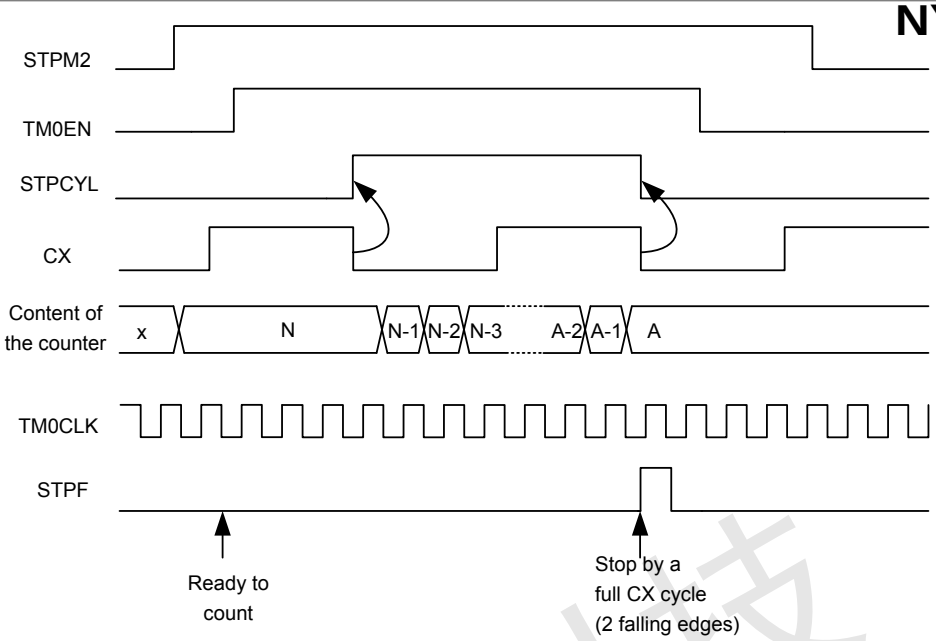

Figure 11-4: Timing of the RFC counter controlled by CX full cycle

11.3.4 Timer0 Counting within a Full Timer2 Clock Cycle (STOP Mode 3)

In this mode, Timer2 clock cycle will dominate the operation of the Timer0(8-bit), or Timer0&1(16-bit) counter. At first, users have to swap to mode 3 through register TM0C, load counting data and turn on Timer0, or Timer0&1 and Timer2 for the initial setting. **It strongly recommends users to switch on Timer0 at last.** Instead of falling edge of CX pin, this mode utilizes two falling edges of Timer2 clock to control the enabling of the 8, or 16-bit counter. The CX pin is applied to be the clock input of the counter. Provided the second falling edge of Timer2 occurs, a pulse of signal STPF would be generated to stop the counting procedure. In this case, if the interrupt enable (IEF[2]) is provided, the interrupt is accepted.

Users can read the content of the counter through register TMxD (x=0, 1) by instruction. The timing procedure of this mode is shown as Figure 8-7.

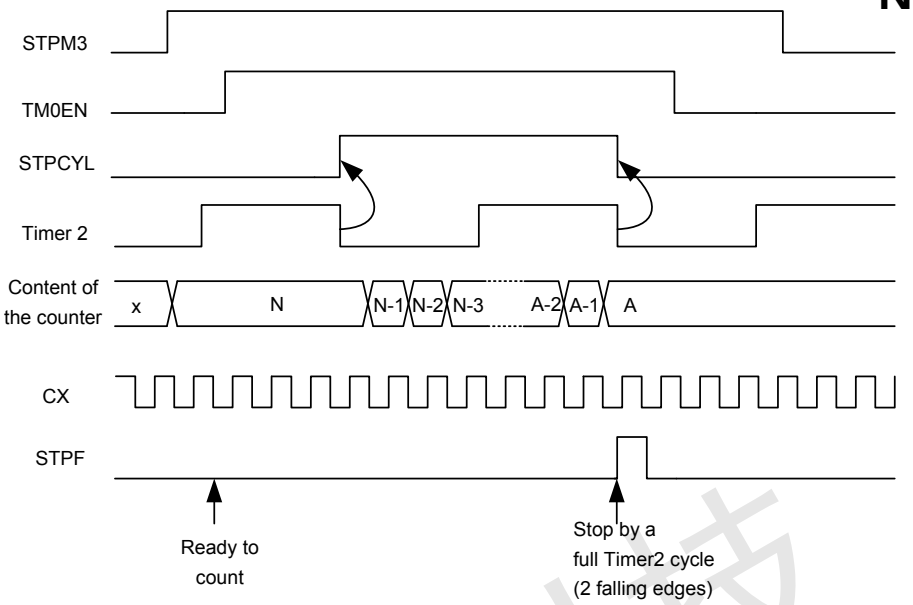

Figure 11-5: Timing of the RFC counter controlled by Timer2 full cycle

11.4 Infrared (IR) Transmitter

The NY8LP10A IC provides an infrared transmit block which is used to send infrared signal. Users can select the frequency of IR carrier, low/high carrier and enable/disable the IR output by option. The IR low/high carrier means that if users option the IR low carrier, the IR output port sends infrared signal when the IR data output register value is low, and vice versa.

The control register IRC is to program its output register and port configuration. If IR option select high(drive) carrier, the initial value of IR data output register bit0 (DO) is 0, and vice versa. Besides, the IR counter can be cleared by writing 0 to IRC[3], and which is always read as high.

Addr.	Name	R/W	Bit	Data	Description	Default
$2F	IRC	R/W	[0]	1/0	IR data output register	H/L (by option)
			[1]	1/0	CMOS/Open-Drain of IR output	CMOS
			[2]	1/0	IR Enable/Disable	Disable
		W	[3]	0	Write 0 to initial IR counter (read as high)	x

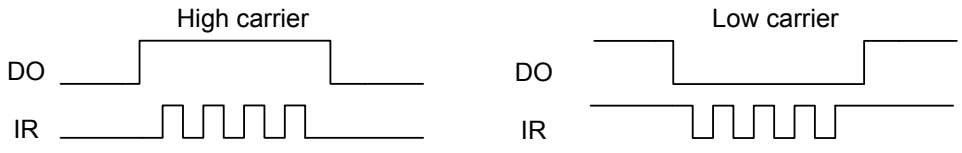

Figure 11-6: Timing of the IR with CMOS output

12. ELECTRICAL CHARACTERISTICS

12.1 Absolute Maximum Rating

Symbol	Parameter	Rated Value	Unit
VDD - VSS	Supply voltage	-0.5 ~ +4.0	V
Vin	Input voltage	VSS–0.3V ~ VDD+0.3	V
Top	Operating Temperature	0 ~ +70	°C
Tst	Storage Temperature	-25 ~ +85	°C

12.2 DC Characteristics

Symbol	Parameter		VDD	Min.	Typ.	Max.	Unit	Test Condition
V_{DD}	Operating voltage			1.1	1.5	3.6	V	F_{CPU} = 500KHz
				2.0	3	3.6		F_{CPU} = 4MHz
I_{HALT}	Supply Current	Halt mode	1.5		0.1	0.5	uA	Sleep, no load
			3		0.1	0.5		
I_{SB1}		Standby mode1	1.5		1		uA	CPU off, IOSC32KHz on, LCD off, Reg off, no load
			3		2			
I_{SB2}		Standby mode2	1.5		1.5		uA	CPU off, IOSC32KHz on, LCD on, Reg off, no load
			3		2			
I_{SB3}		Standby mode3	1.5		4		uA	CPU off, IOSC32KHz on, LCD on, Reg on, no load
			3		5			
I_{SL}		Slow mode	1.5		6		uA	F_{CPU} = IOSC32KHz, no load
			3		15			
I_{OP}		Normal mode	3		1.0		mA	F_{CPU} = 4MHz, no load
I_{IH}	Input current (Internal pull-low)	Weak (1M ohms)	1.5		1		uA	V_{IN} = VDD
			3		3			
		Strong (100K ohms)	1.5		5			
			3		30			
I_{OH}	Output high current (PA/B, SEG/COM@LED mode)		1.5		-2		mA	V_{OH} = 1.0V
			3		-9			V_{OH} = 2.0V
I_{OL1}	Output low current (PA/B)		1.5		4		mA	V_{OL} = 0.5V
			3		18			V_{OL} = 1.0V
I_{OL2}	Output low current (SEG/COM@LED mode)		1.5		2		mA	V_{OL} = 0.5V
			3		9			V_{OL} = 1.0V
ΔF/F	Frequency deviation by voltage drop(500KHz)		1.5		-0.5		%	Fosc(1.5V) - Fosc(1.2V) / Fosc(1.5v)
	Frequency deviation by voltage drop(4MHz)		3		-0.5			Fosc(3.0V) - Fosc(2.4V) / Fosc(3.0v)
ΔF/F	Frequency lot deviation (500KHz)		1.5	-1.5		1.5	%	Fosc(1.5V) - 500KHz / 500KHz
	Frequency lot deviation (4MHz)		3	-1.5		1.5	%	Fosc(3.0v) - 4MHz / 4MHz

Symbol	Parameter	VDD	Min.	Typ.	Max.	Unit	Test Condition
Fosc	Oscillation Frequency	--	0.48	0.5	0.52	MHz	V_{DD} = 1.1~3.6V
			1.95	2	2.05		
			3.9	4	4.1		

13. APPLICATION CIRCUITS

13.1 Application Circuit

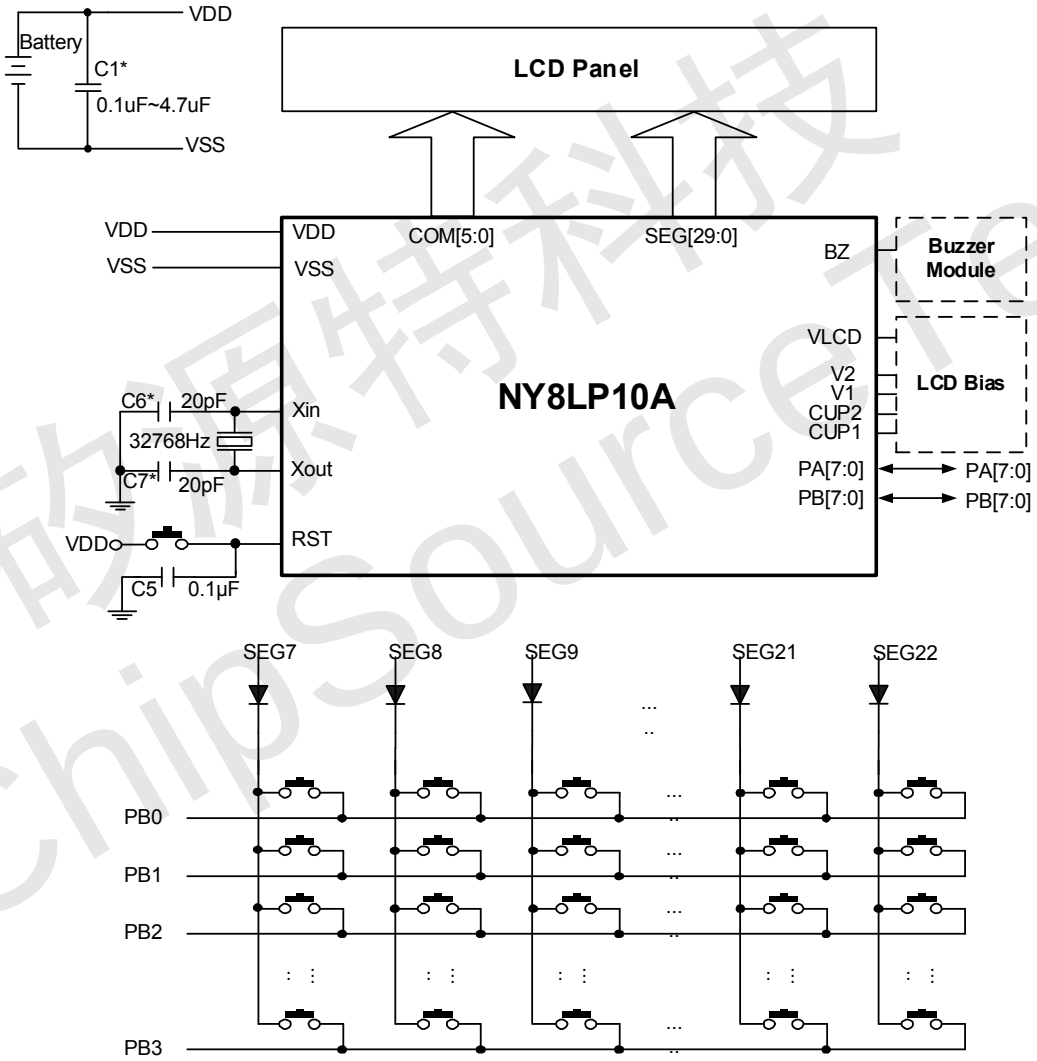

Fig.13-1: The Application Circuits with LCD

PCB Layout Guidelines:

1. VDD must be connected to power input port directly, not the branch of each other.

2. VLCD should be higher than or equal to VDD, otherwise will cause large current.

3. VSS must be connected to ground input directly, not the branch of each other.

4. Capacitor (used for XTAL32K) is proposed to be 12~20 pF.

5. C1 is suggested 0.1uF~4.7uF.

13.2 LCD Bias (VDD for VLCD/V2/V1 or internal Vreg for V1)

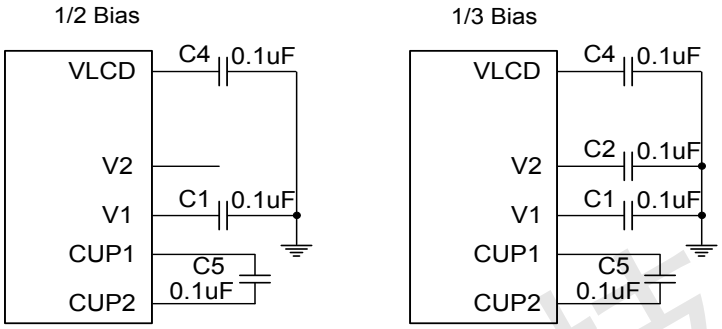

Fig.13-2: The diagram of LCD Bias based on VDD or internal Vreg

14. DIE PAD DIAGRAM